

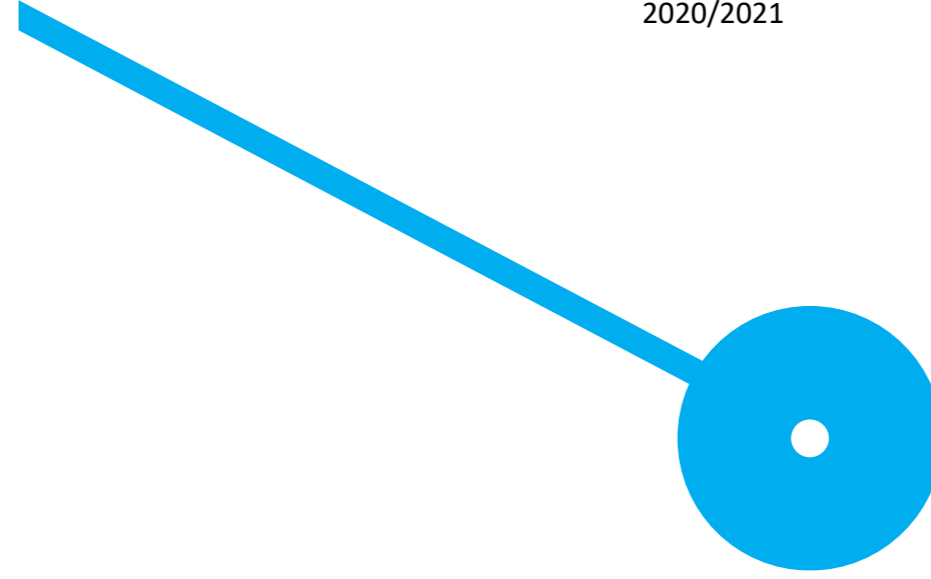
Partilha de Informação 4.0 – Uma abordagem
baseada em International Data Spaces
Pedro Alberto Fernandes Pinheiro

2020/2021

Pedro Alberto Fernandes Pinheiro. Partilha de Informação 4.0 – Uma abordagem
baseada em International Data Spaces

Partilha de Informação 4.0 – Uma
abordagem baseada em
International Data Spaces
Pedro Alberto Fernandes Pinheiro

2020/2021





Partilha de Informação 4.0 – Uma abordagem baseada em International Data Spaces

Pedro Alberto Fernandes Pinheiro

Prof. Doutor Cristóvão Dinis Polido Sousa

Agradecimentos

Começo por agradecer aos meus pais, pela confiança, por me terem apoiado e ajudado durante todo o percurso académico e pessoal. Um grande obrigado, aos meus amigos pela amizade, pelas experiências e por me terem ajudado. Um especial agradecimento ao meu orientador o Prof. Doutor Cristóvão Dinis Polido Sousa pela disponibilidade, confiança, suporte, tema proposto e pela orientação e conselhos dados que me ajudaram ao longo deste projeto. Um obrigado ao Eng. Cesar Toscano e ao Eng. Narciso Sobral, pela disposição e apoio prestado durante o desenvolvimento do projeto. Agradeço ao INESC TEC, pela oportunidade dada. Agradeço a IDSA, em especial à equipa por detrás do DataSpace Connector, pelo trabalho realizado e pelo suporte. Por último, agradeço à Escola Superior de Tecnologia e Gestão, por me acolher, fazer-me evoluir e ajudar-me ao longo destes árduos 5 anos.

Resumo

Nos últimos anos, a necessidade de partilhar informação tem vindo a aumentar, devido ao crescimento exponencial do número de empresas que estão a apostar na transformação digital. As empresas começaram a tratar os dados como um novo ativo económico e passaram a ter um papel importante na tomada de decisão das empresas. A necessidade de partilhar informação, obrigou as empresas a adotarem abordagens que permitissem a partilha de informação quer num contexto horizontal quer vertical. A partilha de informação num contexto vertical consiste na troca de dados entre os vários níveis de tomada de decisão de uma empresa, por outro lado, a partilha de informação num contexto horizontal, representa a troca de dados entre empresas pertencentes ao mesmo ecossistema. Contudo, este último contexto conta com um conjunto de problemas relacionados com a interoperabilidade, flexibilidade e compatibilidade entre sistemas. Este documento descreve os International Data Spaces, como um modelo disruptivo para a partilha de dados interempresarial, com o objetivo de solucionar este conjunto de problemas. Este trabalho será avaliado no âmbito de plataformas marketplace.

Palavras-chave: International Data Spaces, Transformação Digital, Partilha de Informação, Integração Horizontal

Abstract

In the last years, the need to share information has kept growing, due to the exponential growth on the number of organizations that are betting on the digital transformation. The organizations started to treat the data as a new economic asset, by using them to help the process of decision making. The need to share information, lead the companies to adopt new approaches that allow them to share information whether in a horizontal or vertical context. Sharing information in a vertical context consists of sharing data between the different levels of decision making inside a organization. On the other hand, sharing information in a horizontal context represents sharing the information between organizations within the same ecosystem. However, this last context contains a set of problems that are related to the interoperability, flexibility and compatibility within systems. This document, describes the International Data Spaces, as a disruptive model to share information between enterprises, with the goal to solve this set of problems. This work will be evaluated within the scope of marketplace platforms.

Keywords: International Data Spaces, Digital Transformation, Sharing Information, Horizontal Integration

Conteúdo

Agradecimentos	i
Resumo	ii
Abstract	iii
Lista de Figuras	vii
Siglas e Abreviaturas	x
1 Introdução	1
1.1 Objetivos	3
1.2 Principais Resultados	3
1.3 Estrutura do Documento	4
2 Paradigmas para partilha de informação no contexto interorganizacional	6
2.1 Middleware	6
2.2 Espaços de Dados	8
2.2.1 Espaços de Dados Semânticos	9
2.2.2 Espaços de Dados (Abertos) Ligados	10
3 Partilha de Informação 4.0	12

3.1	RAMI 4.0	13
3.2	International Data Spaces	15
3.2.1	Modelo da arquitetura de referência do IDS (IDS-RAM)	17
3.2.2	IDS-RAM vs RAMI 4.0	28
3.2.3	Implementações	29
3.3	GAIA-X	33
3.4	Desafios para as PMEs	36
4	Arquitetura IDS para a partilha de informação interorganizacional	38
4.1	Metodologia	39
4.2	Protótipo IDS	41
4.2.1	Integração com Apache Camel	42
4.2.2	Detalhe da Arquitetura	44
4.2.3	Interações	49
5	Caso de Uso	51
5.1	Descrição do Problema	52
5.2	Arquitetura da Solução	53
6	Conclusões e Trabalho Futuro	58
6.1	Reflexão Crítica	59
6.2	Trabalho Futuro	59
	Bibliografia	60

A	Guia de Configurações	64
A.1	Pré-requisitos	64
A.2	Configurações do Provider	65
A.3	Configurações do Consumer	74

Lista de Figuras

1	Exemplo de um Eletronic Data Interchange	7
2	Exemplo de Dados Ligados entre si	10
3	RAMI 4.0 – Modelo de Arquitetura de Referência para a Indústria 4.0.	13
4	Diagrama dos participantes presentes no International Data Space.	17
5	Estrutura do IDS-RAM.	18
6	Representações do Modelo de Informação.	21
7	Concern Hexagon do Digital Resource.	22
8	Arquitetura de referência de um IDS Connector.	25
9	Relações entre as entidades do IDS Connector.	26
10	Principais interações entre os IDS Connectors.	27
11	Context Broker.	32
12	Exemplo de integração de um FIWARE GE com um IDS Connector.	32
13	Arquitetura do GAIA-X.	35
14	Plano de Trabalho.	39
15	Visão genérica do Serviço Camel no contexto IDS.	43

16	Visão geral da Arquitetura do Protótipo.	43
17	Diagrama de Sequência da arquitetura do protótipo.	49
18	Arquitetura do terceiro objetivo do projeto GREAT.	53
19	Arquitetura da solução apresentada para o terceiro objetivo.	54
20	Exemplo de uma mensagem no MQTT Box.	54
21	Exemplo de uma mensagem recebida no Camel Service Provider.	55
22	Exemplo de uma mensagem recebida no Camel Service Consumer.	56
23	Exemplo de uma mensagem recebida no NodeRed.	56

Exemplos de Código

1	Exemplo de uma Rota no CSP.	46
2	Exemplo de uma Rota no CSC.	48

Siglas e Abreviaturas

API - Application Programming Interface

B2B - Business-to-Business

BPMN - Business Process Model Notation

COP - Community of Practice

CPS - Cyber-Physical Systems

CRM - Customer Relationship Management

CSC - Camel Service Consumer

CSP - Camel Service Provider

EAI - Enterprise Application Integrations

EDI - Eletronic Data Interchange

ERP - Enterprise Resource Planning

EXIC - External IDS Connector

HTTP - Hypertext Transfer Protocol

IDS - International Data Space

IDS-RAM - IDS Reference Architecture Model

IDSA - International Data Space Association

IEC - International Electrotechnical Commission

IIRA - Industrial Internet Reference Architecture

INIC - Internal IDS Connector

IoT - Internet of Things

IT - Information Technology

KOS - Knowledge Organization System

MQTT - Message Queuing Telemetry Transport

OWL - Ontology Web Language

RAMI - Reference Architecture Model for Industry

RDF - Resource Description Framework

RDF-S - Resource Description Framework Schema

REST - Representational State Transfer

SCM - Supply Chain Management

SoC - Separation of Concerns

URI - Uniform Resource Identifier

VMI - Vendor Management Inventory

W3C - World Wide Web Consortium

XML - Extensible Markup Language

Capítulo 1

Introdução

Atualmente, a capacidade de gestão e partilha de informação é considerado um fator crítico de sucesso na definição de uma estratégia de gestão organizacional. Na verdade, o principal fator impulsionador da quarta revolução industrial foi exatamente a informação. A capacidade de recolha e processamento de grandes volumes de dados de forma eficiente e eficaz, constitui um ativo crucial no processo de tomada de decisão, na promoção da inovação e na criação de valor. Consequentemente, tem-se observado [1], um enorme crescimento de iniciativas para a digitalização e automação, com vista à flexibilização do processo produtivo, como forma de satisfazer e lidar com a dinâmica das necessidades do mercado. Contudo, muitas das estratégias adotadas pelas organizações nas suas iniciativas de digitalização são ad-hoc [1], condicionando a escalabilidade de processos e soluções, e dificultando a interoperabilidade e integração intra e interorganizacional. As organizações já não estão, ou pelo menos não podem estar, fechadas em si próprias e a informação necessária para o processo de tomada de decisão encontra-se para além das fronteiras das organizações e da própria cadeia de logística. O paradigma da Indústria 4.0, oferece um conjunto de práticas, políticas e standards num modelo de referência, cujo objetivo é uniformizar e acelerar o processo de transformação digital das empresas, garantindo maior flexibilização, integração e produtividade. Concretamente, os grandes objetivos da Indústria 4.0 consistem na integração de toda a cadeia de valor numa perspetiva horizontal e vertical, envolvendo processos, pessoas e máquinas [1]. A integração vertical consiste na agregação de empresas que pertencem à mesma cadeia de valor de um produto. As empresas verticalmente integradas têm a possibilidade de explorar os seus produtos em todos os níveis da sua cadeia de valor [2]. A integração horizontal consiste na junção de organizações que pertencem ao mesmo setor, de forma a adquirirem um

maior controlo sobre o mercado. Normalmente, esta integração ocorre entre empresas concorrentes [2]. O desafio é, portanto, bastante complexo e revisita problemas antigos como a interoperabilidade dos dados, a uniformização de processos e arquiteturas de suporte à partilha de informação.

Para além das barreiras socioculturais [3], as iniciativas de digitalização enquadradas com os princípios, boas práticas e modelos da Indústria 4.0, requerem a adoção de arquiteturas disruptivas e escaláveis, suportadas por tecnologias emergentes [1]. A este nível mais tecnológico, as organizações têm vindo a debater-se com vários problemas, desde incompatibilidade de sistemas de hardware e software, produtos desenvolvidos em diferentes linguagens de programação e a falta de standards e definições [4]. Este último será, porventura, um dos mais importantes. Mas, se ao nível da integração vertical as arquiteturas IoT e Cloud, têm garantido o fluxo de informação desde o chão-de-fábrica até aos diferentes níveis de tomada de decisão, os desafios inerentes à partilha de informação em contexto interempresarial são mais complexos [4]. Da integração horizontal, emerge a necessidade de maior cuidado na forma como os dados são tratados e partilhados, debatendo-se diversas questões e problemas relacionados com a integração dos dados para que as empresas consigam partilhar informação de forma eficiente, segura, sem comprometer o seu conteúdo, nem perder o controlo sobre as transações [5], ou seja, mantendo a propriedade dos dados.

Apesar da existência de modelos de referência, como por exemplo o RAMI 4.0 e o IIRA, escasseiam frameworks que permitam implementar, em termos práticos, soluções assentes nos pressupostos dos modelos de referência existentes. Surgem, no entanto, algumas iniciativas, tais como os IDS ¹ ou o GAIA-X ², como abordagens mais paradigmáticas e normalizadas à implementação de soluções escaláveis para a integração horizontal na indústria, como por exemplo, os Marketplaces. É no contexto do estudo e avaliação da exequibilidade deste novo tipo de soluções que surge este trabalho.

¹<https://internationaldataspaces.org/>

²<https://www.data-infrastructure.eu/>

1.1 Objetivos

Neste trabalho discute-se o problema da partilha de informação e da interoperabilidade dos dados no contexto interempresarial. Nesse sentido, são estudadas novas abordagens com o potencial para responder os principais desafios relacionados com a troca de dados interempresarial como a segurança, a interoperabilidade, a confiança e a flexibilidade.

Concretamente pretende-se:

1. Validar os International Data Spaces para a partilha de informação entre empresas (Integração Horizontal);
2. Desenhar soluções onde são utilizados Marketplaces orientados a IDS de forma a possibilitarem a troca de dados segura e em tempo real;
3. Validar uma arquitetura baseada em IDS, explorando o seu componente principal, os IDS Connectors, através de um caso de uso no âmbito dos Marketplaces.

Após o cumprimento dos objetivos, é esperado obter uma solução no contexto dos Marketplaces suportados pelos International Data Spaces e um protótipo de um IDS Connector validado no contexto do projeto europeu GREAT, um piloto do projeto Market 4.0 ³.

1.2 Principais Resultados

Perante os objetivos apresentados anteriormente, o trabalho motivou um estudo prévio das diversas abordagens utilizadas para a partilha de informação interorganizacional, nomeadamente os International Data Spaces. Através deste estudo foi possível compreender o seu modelo de referência, os seus componentes principais, entender o seu funcionamento e compreender as suas vantagens e desvantagens face às necessidades industriais no domínio da partilha de dados interorganizacionais.

Após o estudo efetuado, foi desenhada uma arquitetura IDS aplicada a Marketplaces, onde existe troca de mensagens entre sistemas heterogêneos. A arquitetura desenhada utiliza os

³<http://market40.eu/>

IDS Connectors (o principal componente da arquitetura de referência do IDS), e um conjunto de serviços que possibilitam a integração desses IDS Connectors com a realidade de cada empresa (e.g. ERP, CRM, SCM).

Por fim, a solução apresentada foi aplicada a um caso de uso, no âmbito do projeto GREAT ⁴ onde foi verificada a exequibilidade da troca de dados, em tempo real, na rede da arquitetura desenvolvida e que se encontra alinhada com os pressupostos dos IDS. Garante, para além da comunicação ponto-a-ponto entre dois ou mais IDS Connectors, a integração entre os IDS Connectors e os sistemas de informação internos da organização.

1.3 Estrutura do Documento

O documento encontra-se dividido em 6 capítulos, cada capítulo aborda um tema relevante para a dissertação.

- **Capítulo 1 - Introdução:** é apresentada a motivação e o contexto do trabalho, bem como os seus objetivos, os principais resultados e como é estruturado o documento.
- **Capítulo 2 - Paradigmas para partilha de informação no contexto interorganizacional:** são apresentadas as diversas abordagens que visam resolver os problemas relacionados com a partilha de dados interempresarial, como a interoperabilidade dos dados, a flexibilidade e a compatibilidade entre sistemas.
- **Capítulo 3 - Partilha de Informação 4.0:** são identificados os vários modelos de referência para a partilha de informação, em especial o modelo de arquitetura de referência do IDS, e um conjunto de desafios e barreiras que impedem a adoção dos IDSs pelas pequenas e médias empresas.
- **Capítulo 4 - Arquitetura IDS para partilha de informação interorganizacional:** é feita uma descrição dos desenvolvimentos realizados ao longo do trabalho e é apresentada a metodologia utilizada.
- **Capítulo 5 - Caso de Uso:** é apresentado o projeto e problema onde se enquadra o trabalho desenvolvido, e também o resultado da implementação da solução.

⁴<https://www.inesctec.pt/pt/projetos/great>

- **Capítulo 6 - Conclusões e Trabalho Futuro:** é efetuada uma conclusão de todo o trabalho realizado, assim como a sua reflexão crítica e trabalhos futuros.

No final do documento encontra-se um apêndice, onde são definidas as configurações necessárias para iniciar a solução apresentada ao longo deste documento.

Capítulo 2

Paradigmas para partilha de informação no contexto interorganizacional

A partilha de informação interorganizacional é um tema de renovado interesse no contexto da quarta revolução industrial [1]. Contudo, o tema tem vindo a ser debatido há várias décadas, com várias soluções e abordagens, que marcaram de forma significativa a maneira como as organizações se relacionam. Para melhor se compreender os desafios atuais, é relevante discutir os diferentes paradigmas para a partilha de informação num contexto interorganizacional utilizados ao longo dos últimos anos.

2.1 Middleware

Durante os anos 70, as empresas procuraram alternativas que viessem substituir o envio de documentos em formato papel como forma de comunicação, que para além de ser demorada, proporcionava a ocorrência de erros e custos significativos [6]. Assim, surgiu o Eletronic Data Interchange(EDI) [7], que veio substituir a partilha de documentos em formato papel, para a partilha de informação digital (maioritariamente documentos relacionados com compras e vendas), dando-se início a uma nova era de comércio eletrónico do tipo Business-to-Business(B2B) [8]. Esta transição, trouxe vantagens para as empresas como a redução de custos no processamento destes documentos, automação nas operações, simplicidade e

rapidez nos processos de transferências [9]. O EDI, oferece uma forma de comunicação estruturada através de redes privadas entre empresas. Para assegurar a compatibilidade entre pedidos, o EDI, utiliza uma série de padrões para impor regras na composição das mensagens trocadas [7].

A implementação do EDI além de ser complexa, é necessário realizar um grande investimento por parte da empresa, o que torna bastante difícil a sua implementação em pequenas e médias empresas. Contudo, surgiram alternativas ao EDI, onde a sua implementação é mais simples e com menor custo para as pequenas e médias empresas [9], como o Web EDI [10] e o e-business XML [11]. Ainda hoje existem empresas que mantêm estas abordagens para a troca eletrónica de dados (documentos de negócio).

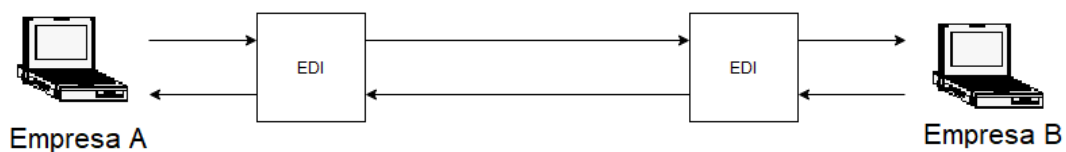


Figura 1: Exemplo de um Eletronic Data Interchange

O Web EDI, é uma evolução do EDI que utiliza a internet para estabelecer comunicação entre parceiros, através de portais acedidos por web browsers, para a troca de um número limitado de mensagens [10]. Esta alteração permite incluir as pequenas e médias empresas no processo. Nesta abordagem [10], os utilizadores, para submeterem a informação que anteriormente se encontrava em documentos no formato de papel, preenchem formulários web, que podem já conter alguma informação definida pelos parceiros de negócio através de regras.

Apesar do custo da implementação de um Web EDI ser menor que o EDI tradicional, muitas empresas continuaram a não apostar em EDI. Isso deve-se ao facto de ser necessário definir acordos baseados em padrões EDI [11] entre as empresas participantes, o que obriga às empresas a adotarem os seus sistemas para conseguirem trocar mensagens entre si, dificultando assim a entrada de pequenas e médias empresas num mercado global. Para solucionar este problema, surgiu uma abordagem baseada em XML, que permite definir mensagens com base no padrão XML, facilitando a definição de mensagens e implementação tecnológica.

Um exemplo de utilização de EDI no contexto da troca de informação interorganizacional é o Vendor Management Inventory (VMI) [12]. O VMI é um processo responsável pela gestão de stock, onde o fornecedor envia a matéria-prima ao cliente, automaticamente, com base na movimentação de stock do cliente e no acordo base de fornecimento de matéria-prima.

2.2 Espaços de Dados

As abordagens baseadas em Middleware necessitam de um grande esforço para a estruturação de mensagens e para a implementação de sistemas, de modo a permitir o envio e receção segura de informação referente a transações comerciais. Surgem, entretanto, abordagens baseadas em Espaços de Dados, que se focam no conteúdo e não na troca de mensagens. Inicialmente, estas abordagens surgiram no sentido de colmatar as necessidades intraorganizacionais relacionadas com atividades de gestão de informação para o apoio à tomada de decisão, baseadas em conhecimento.

Um Espaço de Dados, consiste numa rede de diversas fontes de dados, que estão relacionadas entre si, com a possibilidade de adicionar ou melhorar funcionalidades com o passar do tempo seguindo uma abordagem "pay-as-you-go" [13]. Conceptualmente podemos definir estes espaços como uma estrutura geométrica que é capaz de representar conhecimento através de um conjunto de domínios que por sua vez são representados por um conjunto de propriedades [14]. Os Espaços de Dados, dispõem de propriedades que resolvem alguns dos problemas relacionados com a integração de dados, como a heterogeneidade das fontes de dados e da sua estruturação.

A maneira de como são definidas e aplicadas regras, aos dados e às suas relações, vai depender do tipo de Espaço de Dados em questão. Um exemplo disso é o Espaço de Dados Partilhado, que consiste num modelo focado na troca de dados entre clientes distribuídos, de modo a permitir a escalabilidade, segurança e eficiência na propagação de updates. As aplicações presentes num Espaço de Dados Partilhado podem sair e entrar no espaço dinamicamente, para aceder aos dados [15].

2.2.1 Espaços de Dados Semânticos

Os Espaços de Dados Semânticos [16] surgem no sentido de estender os Espaços de Dados em dois aspetos fundamentais:

1. A partilha de informação baseada no conteúdo para contextos intra e interorganizacionais;
2. Fornecer uma estrutura comum de organização de informação, facilitando a interpretabilidade e recuperação da informação.

Tipicamente, os Espaços de Dados Semânticos usam Sistemas de Organização de Conhecimento (KOS) [17] para a estruturação de informação e motores de inferência [16] para apoiar a recuperação da informação. O formalismo da representação de informação assenta em standards baseados em XML [18] como Resource Description Framework Schema (RDF-S) e Ontology Web Language (OWL). O RDF-S [19] expande o vocabulário básico de RDF, com o objetivo de fornecer mecanismos para a criação de classes e propriedades sobre os dados, num grafo RDF. A OWL [20] é construída no topo de RDF-S e adiciona vocabulário para descrever melhor as classes, propriedades e relações entre elas, com a intenção de facilitar a representação de ontologias e o processamento nas aplicações.

Recorrentemente, os Espaços de Dados Semânticos são considerados "Espaços de Dados Partilhados", uma vez que um dos grandes objetivos da abordagem semântica aos Espaços de Dados é fomentar a partilha de informação. Este tipo de tecnologia tem sido usado na implementação de sistemas de gestão do conhecimento e plataformas digitais para suportar as Comunidades de Prática (COP) [21], devido à sua escalabilidade, segurança e eficiência na troca de informação, entre os participantes.

A stack tecnológica na base da estruturação dos Espaços de Dados Semânticos ou Partilhados, é a que caracteriza a Web Semântica [22]. Da abordagem semântica associada à web, emerge o conceito de Dados Ligados entre si, Dados Abertos e um novo tipo de Espaço de Dados, os Espaços de Dados Ligados.

2.2.2 Espaços de Dados (Abertos) Ligados

Os dados fornecidos pelos participantes dos Espaços de Dados, muitas das vezes possuem os seus próprios formatos e standards, com o objetivo de definir regras para que esses dados possam ser processados pelos próprios sistemas de documentos. Devido à utilização desses standards, torna-se muito difícil partilhar e relacionar esses dados entre os diversos participantes do Espaço de Dados, pois cria uma enorme incompatibilidade entre os vários sistemas. O conceito de Dados Ligados veio corrigir essa diversidade que existe nos formatos e sistemas de documentos, ao utilizar standards comuns entre os participantes. Podemos definir Dados Ligados como um conjunto de princípios, que tem como objetivo publicar e estruturar os dados, de forma que possam ser acedidos por qualquer máquina [22]. Isso só é possível, através do uso da Resource Description Framework (RDF) para formatar os documentos, e de Uniform Resource Identifiers (URIs) para identificar toda a informação que pode ser acedida, desde documentos até pessoas e objetos [22]. Os Dados Ligados contam com os seguintes princípios:

1. Os dados têm de ser identificados por Uniform Resource Identifiers (URIs);
2. Os dados identificados por URIs devem ser pesquisados através do HyperText Transfer Protocol (HTTP);
3. Quando um URI é pesquisado, os dados retornados têm de seguir os padrões da Resource Description Framework (RDF);
4. Os URIs estão relacionados entre si, através de hyperlinks (RDF Links).

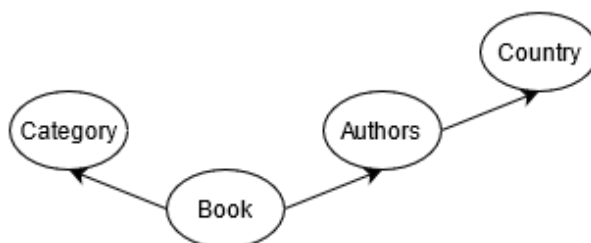


Figura 2: Exemplo de Dados Ligados entre si

Os Espaços de Dados Ligados combinam os princípios de Dados Ligados com Espaços de Dados, de forma que seja possível relacionar os dados entre as diversas fontes de dados (Figura 2). Essas relações permitem a criação de plataformas capazes de executar pesquisas

que retornam todos os dados necessários, mesmo se estes pertencerem a diferentes fontes de dados [23].

Com o surgimento do conceito de Dados Abertos as organizações começaram a disponibilizar e obter os dados de terceiros, com o intuito de explorar o seu mercado, identificar novas oportunidades de negócio e criar novos produtos e serviços [24]. Podemos definir Dados Abertos como um conjunto de dados disponíveis ao público, onde todos os utilizadores têm acesso total sobre esses mesmos dados, sem qualquer restrição [25]. Através da introdução de Dados Abertos é possível formar um ecossistema para a partilha e criação de novo conhecimento, que pode ser usado em atividades de inovação. Ao aplicarmos o conceito de Dados Abertos, dá-se início a um novo tipo de Espaço de Dados Ligados, o Espaço de Dados Ligados e Abertos [22], semelhante à Internet dos dias de hoje onde a partir de conhecimento conseguimos gerar novo conhecimento.

Os paradigmas apresentados, solucionaram grande parte dos problemas relacionados com a partilha de informação interorganizacional, contudo foram surgindo novos desafios que obrigaram a um "shift" paradigmático no âmbito da partilha de informação, que deu início à atual partilha de informação 4.0.

Capítulo 3

Partilha de Informação 4.0

A inovação tem um papel crucial na revolução industrial, pois é através da inovação que vão surgindo novos avanços industriais e tecnológicos. Estes avanços tecnológicos, e consequente crescimento da digitalização, automação e flexibilização dos processos de produção, criaram a necessidade de explorar novas tecnologias, especificar padrões, arquiteturas e digitalizar os vários processos de produção.

Assim sendo, entramos numa nova revolução industrial, a **Indústria 4.0**, que proporcionou a integração dos processos de produção e de negócio, assim como todos os participantes da cadeia de valor (clientes e fornecedores), numa perspetiva horizontal e vertical [1]. A Indústria 4.0, visa explorar o potencial dos avanços tecnológicos e de novos conceitos, com o objetivo de dinamizar toda a produção de acordo com os requisitos do mercado, através da:

- Criação de Cyber-Physical Systems (CPS) ¹, que se encontram constantemente a partilhar informação entre si, com o objetivo de identificar, monitorizar e otimizar os processos de produção;
- Criação de arquiteturas baseadas em Internet of Things (IoT) ²;
- O processamento de grandes quantidades de dados obtidos através de processos, máquinas e produtos;
- A introdução de Smart Factories ³.

¹<https://www.i-scoop.eu/industry-4-0/32-cyber-physical-systems-the-building-blocks-of-industry-4-0/>

²<https://www.i-scoop.eu/internet-of-things-iot/>

³<https://www.i-scoop.eu/industry-4-0/smart-factory-scaling/>

O desafio é, portanto, bastante complexo e revisita problemas antigos como a interoperabilidade dos dados, a uniformização de processos e arquiteturas de suporte à partilha de informação. A criação de diversos modelos de referência, padrões e conceitos pretende responder a esses problemas.

3.1 RAMI 4.0

Com a introdução da Indústria 4.0 e de novos conceitos, grande parte das empresas optaram por continuar a utilizar os seus sistemas e os seus modelos de dados e negócio. Consequentemente, foram surgindo vários desafios relacionados com a utilização desses mesmos modelos de dados e negócio. O RAMI 4.0 (Reference Architecture Model for Industry 4.0) foi desenvolvido com o objetivo de resolver os vários problemas relacionados com a aplicabilidade dos pressupostos inerentes à quarta revolução industrial, ao oferecer às empresas uma estrutura base para a criação de novos produtos e de modelos de negócio [26], tendo por base a reutilização dos standards existentes na indústria.

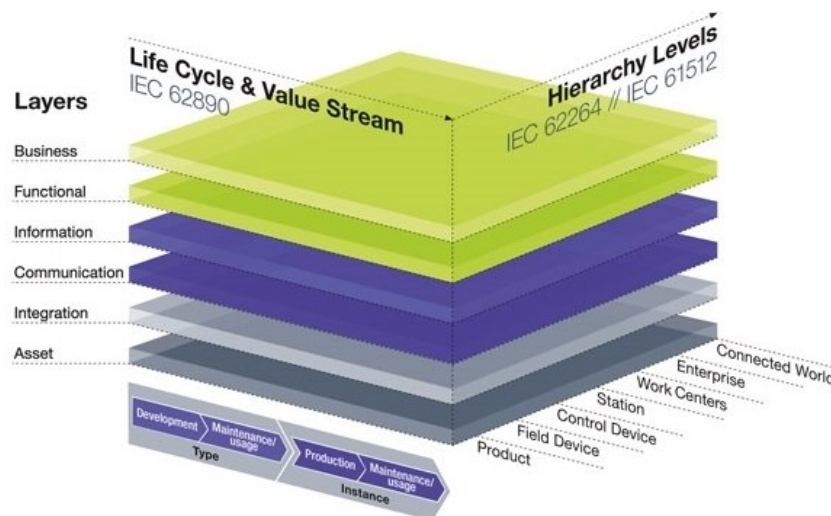


Figura 3: RAMI 4.0 – Modelo de Arquitetura de Referência para a Indústria 4.0 [26].

O RAMI 4.0 é constituído por três dimensões, que têm como principal objetivo, combinar todos os elementos e componentes TI ⁴ dentro de uma "Camada" e um "Modelo do Ciclo de Vida", de forma a certificar-se que todos os participantes envolvidos consigam comunicar entre si. O modelo oferece um linguagem comum a todos os participantes, de modo a aplicarem os conceitos da Indústria 4.0 [27].

⁴Tecnologia da Informação

O eixo vertical "Camadas"(Layers), é responsável por descrever os componentes TI, de forma estruturada, através das seguintes camadas:

- A **Camada de Negócio (Business Layer)**, inclui todo o modelo de negócio e faz a ligação entre os diversos processos de negócio;
- A **Camada Funcional (Functional Layer)**, possui uma descrição formal de todas as funções de uma componente pertencente à Indústria 4.0;
- A **Camada de Informação (Information Layer)**, corresponde à integração e processamento dos dados em informação útil;
- A **Camada de Comunicação (Communication Layer)**, é responsável pela standardização da comunicação entre componentes, através de protocolos e formatos, que contribuem para a sua interoperabilidade. Além disso, oferece serviços para a Camada de Integração;
- A **Camada de Integração (Integration Layer)**, representa a transição entre o mundo real e o mundo da informação;
- A **Camada de Ativos (Asset Layer)**, inclui todos os componentes que pertencem ao mundo real (físicos ou digitais), como sensores, software e máquinas.

O eixo horizontal da esquerda "Ciclo de Vida e Fluxo de Valor"(Life Cycle & Value Stream), representa o ciclo de vida do produto segundo o standard IEC 62890. Neste contexto, um *Type* corresponde a uma ideia, plano ou conceito de um produto ainda não disponível, já uma *Instance* representa um produto já desenvolvido e disponível no mercado. Um *Type* torna-se *Instance* quando o desenho e a prototipagem terminam e o produto começa a ser produzido [28].

Por fim, o eixo horizontal da direita "Níveis de Hierarquia"(Hierarchy Levels), através dos standards IEC 62264 e IEC 61512, faz uma representação de todos os níveis hierárquicos de uma organização e das suas funcionalidades, de maneira a simplificar a explicação de vários processos de produção complexos [29]. O eixo está dividido nas seguintes categorias:

- Os **Mundos Conectados (Connected Worlds)**, descrevem as relações entre as várias componentes;
- A **Organização (Enterprise)**, representa a organização e as suas iniciativas;

- As **Unidades de Trabalho (Work Centers)**, abrangem todas as secções presentes nas linhas de produção;
- A **Estação (Station)**, compreende todos os ativos industriais;
- O **Dispositivo de Controlo (Control Device)**, inclui todos os controladores (exemplo: Controlador Lógico Programável);
- O **Dispositivo de Máquinas (Field Device)**, representa os vários sensores;
- O **Produto (Product)**, corresponde ao produto.

O modelo RAMI 4.0, através das suas dimensões, consegue alcançar os dois tipos de integrações (vertical e horizontal). No entanto, um dos objetivos deste trabalho passa por responder aos desafios impostos pela integração horizontal e por isso é necessário encontrar modelos mais práticos e mais adaptados às pequenas e médias empresas, para implementar arquiteturas de sistemas no âmbito das suas estratégias de transformação digital, de modo a conseguir eliminar os principais problemas relacionados com a partilha de dados interorganizacional. Desta necessidade surge o IDS-RAM. O IDS-RAM é um modelo, que está enquadrado nos International Data Spaces e encontra-se mais adaptado às realidades das organizações, uma vez que vai para além de uma visão conceptual, fornecendo uma framework para implementações de soluções para troca de dados entre empresas. O objetivo do IDS-RAM é focar-se nos principais problemas relacionados com a partilha de dados interorganizacional, e encontra-se alinhado com o RAMI 4.0.

3.2 International Data Spaces

O grande motor da quarta revolução industrial são os dados, e a partilha dos mesmos torna-se crucial para a implementação do conceito de Fábricas Inteligentes (Smart Factories), que é um dos principais objetivos da Indústria 4.0, focando-se na automatização e otimização dos vários processos de produção, tornando-os mais flexíveis e alinhados com as exigências dos clientes [30]. As Fábricas Inteligentes, introduziram uma nova geração de sistemas de produção, que obrigaram à criação de novos modelos capazes de sublinhar os aspetos mais relevantes para o desenho e o processo de desenvolvimento desses sistemas como por exemplo o RAMI 4.0 (abordado anteriormente) [30]. Contudo, como o RAMI 4.0 é um modelo mais conceptual, foi necessário criar novos modelos, mais práticos e específicos, preparados para responder

num nível mais detalhado aos aspetos sublinhados pelo RAMI 4.0. Um desses modelos é o IDS-RAM, proposto pela iniciativa IDS, que visa responder às questões relacionadas com a partilha de dados entre os sistemas das diversas organizações.

A iniciativa International Data Space (IDS), apresenta um Espaço de Dados onde as organizações têm a possibilidade de trocar dados entre si, sem terem de abdicar dos direitos sobre os seus dados. O IDS aponta para a criação de uma rede de dados descentralizada, que abrange todo o ecossistema do negócio e que a partir de um conjunto de padrões, definidos pela International Data Space Association (IDSA), permitem a troca de dados de forma segura, entre os seus participantes [31]. Os participantes do IDS, para acederem ao IDS necessitam de seguir um conjunto de regras, que asseguram a interoperabilidade dos dados e a sua segurança. A forma como as empresas comunicam dentro de um IDS, depende de conectores para se ligarem ao Espaço de Dados, esses conectores possibilitam, que haja um maior controlo sobre o acesso e a utilização dos dados [32]. O IDS difere dos restantes logo na sua génese conceptual assumindo-se como uma "Network-Oriented" e "Trusted-Based", realizando comunicações ponto-a-ponto, cada um com as suas especificidades garantidas através dos conectores, cuja arquitetura assegura a comunicação segura com base nos acordos definidos pelos participantes e pela IDSA [31].

Dentro de um IDS podemos encontrar as seguintes entidades conceptuais:

- O **Dono dos Dados (Data Owner)**, representa a entidade que é titular dos dados presentes no Fornecedor dos Dados;
- O **Fornecedor dos Dados (Data Provider)**, representa a entidade que fornece os dados ao Consumidor de Dados e define um conjunto de regras de acesso e de utilização para esses mesmos dados, com base nas especificações dos seus titulares;
- O **Consumidor dos Dados (Data Consumer)**, corresponde à entidade que obtém os dados fornecidos pelo Fornecedor dos Dados;
- O **Utilizador dos Dados (Data User)**, representa a entidade que utiliza os dados obtidos pelo Consumidor dos Dados;

O seguinte diagrama apresenta as funções das entidades conceptuais, apresentadas anteriormente.

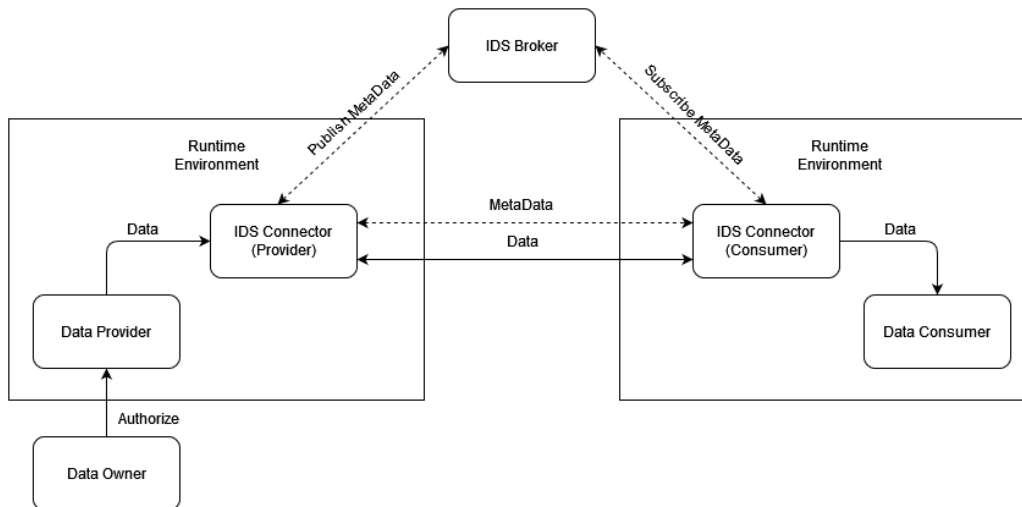


Figura 4: Diagrama dos participantes presentes no International Data Space baseado em [33].

De acordo com o diagrama da figura 4, cada um dos conectores IDS encontra-se numa entidade (organização) distinta, com o seu próprio "Runtime Environment". Poderá existir uma terceira entidade com o papel de IDS Broker, que está responsável pelo envio de notificações entre as restantes entidades. As duas entidades representadas, trocam dados entre si através dos conectores IDS (abordados com mais detalhe na secção 3.2.1.2).

3.2.1 Modelo da arquitetura de referência do IDS (IDS-RAM)

Provavelmente, dois dos aspetos que mais condicionam a implementação e escalabilidade de arquiteturas de interoperabilidade, são os aspetos de confiança/segurança e uniformização (arquiteturas apoiadas em standards e modelos de referência abertos). Os IDSs procuram responder a estes dois desafios, através de um modelo de arquitetura de referência (IDS-RAM). De um ponto de vista mais prático, o modelo tem como objetivo ajudar as organizações a desenhar a sua arquitetura de interoperabilidade interempresarial. Este modelo serve como base para as diversas implementações do IDS e aponta para a criação de uma rede onde os seus participantes possam partilhar dados sensíveis de forma segura. De acordo com a seguinte figura, é possível dividirmos o IDS-RAM em cinco camadas, a Camada de Negócio (Business Layer), a Camada Funcional (Functional Layer), a Camada de Processo (Process Layer), a Camada de Informação (Information Layer) e a Camada de Sistema (System Layer) [31]. Cada camada tem de responder às questões relacionadas com as perspetivas de Segurança (Security), Certificação (Certification) e Governança (Governance).

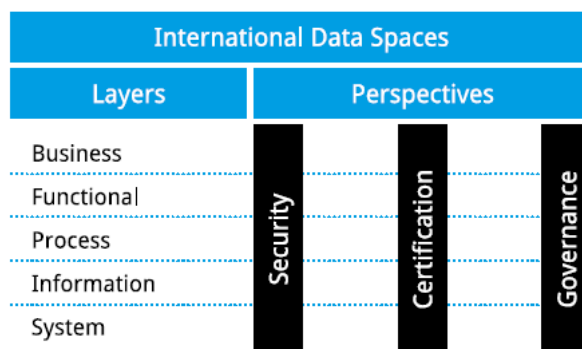


Figura 5: Estrutura do IDS-RAM [31].

A **Camada de Negócio (Business Layer)**, é responsável por definir e categorizar os papéis que os participantes do IDS possam assumir, e também por especificar padrões relacionados com a interação entre esses papéis, assegurando os aspetos relacionados com a segurança e com a governação [31]. Existem quatro tipos de papéis presentes no IDS: o Core Participant, o Intermediary, o Software/Service Provider e o Governance Body. Respetivamente no Core Participant, encontram-se os papéis, necessários para a partilha de dados, como o Data Provider, o Data User, o Data Owner, o Data Consumer e o App Provider. Dentro da categoria, Intermediary situam-se as entidades confiáveis, que estabelecem a ligação entre os Core Participants e fornecem os metadados (exemplo: o Broker Service Provider, a Clearing House, o Identity Provider, a App Store e o Vocabulary Provider). A categoria, Software/Service Provider inclui todas as empresas de TI que fornecem os softwares e serviços aos Core Participants num IDS. Por fim, o Governance Body contém os papéis relacionados com a certificação de todos os participantes no IDS. Cada papel possui um nível específico de certificação descrito com base no Framework for the IDS Certification scheme ⁵. A Camada de Negócio contém os requisitos de negócio, posteriormente especificados na Camada Funcional [31].

A **Camada Funcional (Funcional Layer)**, descreve os requisitos funcionais de uma instância do IDS, independentemente da sua tecnologia. A correta implementação dos requisitos funcionais acaba por criar uma base para a certificação, governação e segurança nas componentes técnicas do IDS. Os requisitos funcionais definidos, podem ser agrupados em seis grupos, de forma a alinharem com os requisitos estratégicos do IDS. Os requisitos estratégicos do IDS são [31]:

- **Confiança (Trust):** A confiança corresponde a um dos pilares do IDS, e por isso todos os participantes têm de ser validados, através de um conjunto de certificados, para poderem

⁵<https://internationaldataspaces.org/wp-content/uploads/IDSA-Strategy-paper-certification-scheme-V.1.pdf>

aceder ao IDS [31];

- **Segurança e Governação dos Dados (Security e Data Sovereignty):** Todos os componentes técnicos do IDS, devem ser avaliados e certificados para garantir a segurança na partilha de dados. Os Data Providers têm a possibilidade de definir políticas de Usage Control para manterem os direitos sobre os dados, durante todo o seu percurso [31];
- **Ecosistema de Dados (Ecosystem of Data):** Os dados são fundamentais para o funcionamento do IDS, e por isso deve ser possível partilhar, descrever, encontrar e interpretar os dados dentro do IDS [31];
- **Interoperabilidade Standardizada (Standardized Interoperability):** A partilha de dados é um aspeto fundamental para o IDS, e por isso é necessário assegurar a interoperabilidade entre os sistemas. O IDS Connector, é a componente central do IDS, utilizado por todos os participantes para assegurar essa interoperabilidade [31];
- **Apps que Adicionam Valor (Value Adding Apps):** A partilha de dados só é possível se os dados forem processados e transformados. O IDS disponibiliza um conjunto de Data Apps, na App Store, que se conectam aos IDS Connectors de maneira a introduzir novos serviços para o processamento de dados antes e depois da sua partilha [31];
- **Mercado de Dados (Data Markets):** O IDS, necessita de integrar vários conceitos relacionados com Data Market para que os Data Providers possam comercializar os seus dados [31].

A **Camada de Processo (Process Layer)**, descreve as interações existentes entre os diversos componentes do IDS, através da criação de processos e subprocessos segundo o Business Process Model Notation (BPMN). A certificação dessas componentes vai depender se estão de acordo ou não com processos envolventes. O IDS conta com três processos principais, o OnBoarding, o Exchanging Data e o Publishing e Using Data Apps, que abrangem todos os participantes definidos na Camada de Negócio. O processo OnBoarding, é composto por vários subprocessos, e tem o objetivo de descrever como é que os participantes se conectam ao IDS. O processo Exchanging Data, possui dois subprocessos, a fim de explicar como é realizada a partilha os dados entre os Data Providers e Data Consumers. E por fim, o processo Publishing e Using Data Apps, descreve como é o processo de certificação das Data Apps é realizado, e como são efetuadas as interações com os IDS Connectors. Estes processos estão diretamente relacionados com a perspetiva de Governação, pois acabam por

descrever interações onde a Governação está presente. De forma a assegurar os aspetos relacionados com a segurança é importante existir monitorização, validação e redesenho caso seja necessário, dos processos existentes [31].

A **Camada de Informação (Information Layer)**, define o Modelo de Informação do IDS (Information Model), que utiliza os padrões usados nos espaços de dados semânticos para descrever os participantes, as suas interações e os recursos partilhados entre eles garantindo a Governação, e as restrições impostas sobre o uso e acesso dos dados. O Modelo de Informação do IDS, funciona como uma linguagem comum entre os participantes e componentes do IDS, para aumentar a compatibilidade e a interoperabilidade entre os seus sistemas, e definir um conjunto de regras de acesso e de controlo nos dados, de forma a ser perceptível por todos e assegurar a perspectiva de Segurança. A certificação de um componente vai depender da sua compatibilidade com o Modelo de Informação do IDS [31].

Na **Camada de Sistema (System Layer)**, os participantes definidos na Camada de Negócio são mapeados para uma arquitetura específica, de modo a cumprir com os requisitos especificados na Camada Funcional, originando assim a estrutura técnica do IDS. Na estrutura técnica do IDS todos os componentes são validados com base nas restantes camadas, de modo a garantir a segurança, governação e certificação dentro do IDS [31].

3.2.1.1 Modelo de Informação do IDS

O Modelo de Informação, presente na Camada de Informação, corresponde a um acordo partilhado pelos participantes e pelas entidades do IDS, com o intuito de promover a interoperabilidade e facilitar a compatibilidade entre as componentes, ao descrever as propriedades essenciais das entidades, dos participantes, dos processos e das componentes presentes no IDS. Através do Modelo de Informação é possível a partilha de dados dentro de uma "Trusted-Network", onde existe a possibilidade de haver o controlo total sobre os dados, por parte dos titulares [31].

É possível descrever o Modelo de Informação através de três tipos de representações, a Representação Conceptual (Conceptual Representation), a Representação Declarativa (Declarative Representation) e a Representação Programática (Programmative Representation).

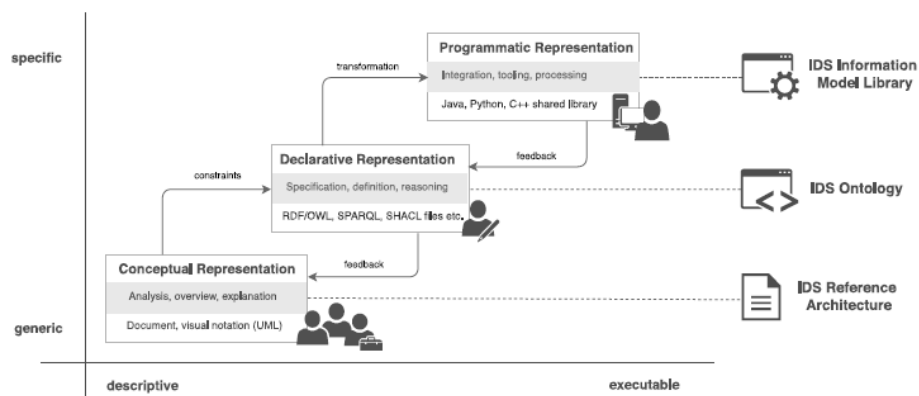


Figura 6: Representações do Modelo de Informação [31].

A **Representação Conceptual (IDS Reference Architecture** ⁶), apresenta uma descrição simples e global dos conceitos presentes, independentemente da sua tecnologia. O objetivo desta representação é permitir a grande parte dos participantes perceber os conceitos, ao disponibilizar informação e ao promover a partilha de conhecimento através de documentação e diagramas [31].

Dentro da Representação Conceptual, encontramos o **Concern Hexagon (C-Hexagon)**, que faz uma representação conceptual das principais preocupações relacionadas com o Digital Resource, que é um recurso único e digital partilhado pelos participantes do IDS, através do principio da Separation of Concerns (SoC) [31]. A SoC, dedica-se a separar um determinado conceito em secções distintas, onde cada secção aborda uma determinada preocupação. Cada preocupação, conta com vários níveis onde se encontra uma vista mais detalhada de um determinado aspeto. A seguinte imagem apresenta o Concern Hexagon do Digital Resource.

⁶<https://internationaldataspaces.org/use/reference-architecture/>

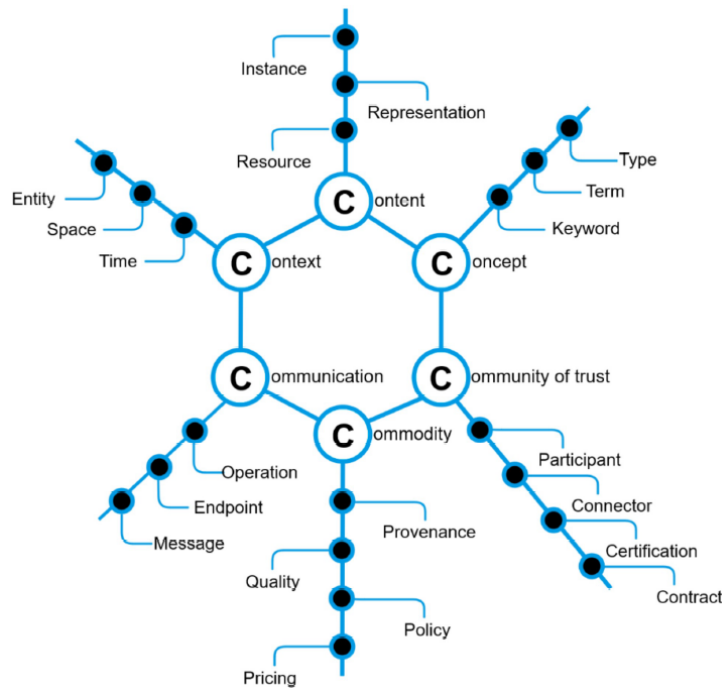


Figura 7: Concern Hexagon do Digital Resource [31].

Através da imagem apresentada, é possível identificar os módulos Contexto (Context), Conteúdo (Content), Conceito (Concept), Comunicação (Communication), Valor (Commodity) e Comunidade de Confiança (Community of Trust). Dos módulos identificados e com base no trabalho, é relevante abordar os seguintes:

- O **Conteúdo (Content)**, compreende tudo o que representa o conteúdo presente nos Resources (Dados e MetaDados). É possível dividir o Conteúdo em três níveis [31]:
 - O Resource, descreve de forma abstrata o seu conteúdo independentemente da sua manifestação física [31];
 - A Representation, adiciona informação estruturada à descrição abstrata realizada no Resource, tornando-a assim mais perceptível [31];
 - A Instance (Artifact), corresponde à materialização de uma Representation, onde são apresentadas propriedades específicas do conteúdo partilhado [31].
- A **Comunicação (Communication)**, aborda tudo o que está relacionado com os meios de comunicação usados para a partilha do conteúdo dentro dos Resources. Dentro do módulo da Comunicação também são encontrados três níveis [31]:
 - A Operation, corresponde a uma transação atômica, que está presente durante a partilha, o processamento, a visualização e na persistência do conteúdo digital [31];

- O Endpoint, representa um ponto onde o conteúdo é partilhado e onde existem interações com um determinado serviço [31];
 - As Messages, descrevem o conteúdo dos dados partilhados entre os conectores. É possível agrupar as Messages em Request Messages, Response Messages e em Notification Messages [31].
- A **Comunidade de Confiança (Community of Trust)**, considera tudo o que é fundamental para partilhar de dados, de forma segura e confiável, entre o Data Owner e o Data Consumer. Na Comunidade de Confiança é possível identificar quatro níveis [31]:
 - O Participant, representa uma entidade ou componente certificada, que está presente no IDS [31];
 - O IDS Connector, é a componente principal do IDS e é usado por todos os participantes pertencentes ao IDS [31];
 - A Certification, determina formalmente e recorrendo a um conjunto de processos de avaliação, se determinado participante cumpre os requisitos para ingressar no IDS [31];
 - O Contract, corresponde a um acordo entre participantes, onde são especificadas as restrições sobre o comportamento e acesso relativamente aos dados partilhados [31].

A **Representação Declarativa (IDS Ontology ⁷)**, utiliza os padrões e vocabulários presentes na stack tecnológica da Web Semântica, para especificar tudo o que foi apresentado na Representação Conceptual, com o objetivo de promover a interoperabilidade e compatibilidade entre máquinas. Através da Representação Declarativa, é originado um modelo de referência base para as várias Representações Programáticas [31].

⁷<https://international-data-spaces-association.github.io/InformationModel/docs/index.html>

A **Representação Programática (IDS Information Model Library** ⁸), oferece aos desenvolvedores a possibilidade de integrar o Modelo de Informação com a infraestrutura de desenvolvimento, através da disponibilização de bibliotecas específicas a cada linguagem. Com a Representação Programática, os desenvolvedores conseguem facilmente criar instâncias do Modelo de Informação, de acordo com a IDS Ontology [31].

3.2.1.2 IDS Connector

O IDS Connector é a componente principal do IDS, usado por todos os participantes para estabelecer a ligação entre as fontes de dados dos Data Providers e as aplicações dos Data Users dentro do IDS. Através do IDS Connector é possível garantir a segurança nas conexões entre participantes confiáveis e definir o controlo de acesso e de utilização dos dados partilhados.

No IDS podemos dividir os conectores em externos e internos. Um IDS Connector Externo (External IDS Connector), permite a troca de dados entre os diferentes participantes do IDS. O IDS, é constituído por uma rede de conectores externos, cada empresa pode se conectar a mais que um IDS Connector Externo e este possuir mais que uma empresa conectada. Um IDS Connector Interno (Internal IDS Connector) semelhante ao IDS Connector Externo, contudo, situa-se dentro de uma rede privada numa empresa. O IDS Connector Interno tem acesso aos dados gerados pela empresa e disponibiliza-os ao IDS Connector Externo [31].

Os IDS Connectors contam com diversos perfis, cada perfil foca-se num determinado aspeto do IDS. Neste trabalho foram estudados os perfis de segurança (Trusted Connector), governação na partilha de dados (DataSpace Connector) e o perfil base (Base Connector). Para facilitar a construção de um IDS Connector, a IDSA apresentou uma arquitetura de referência, apresentada na seguinte figura, que define um conjunto de requisitos mínimos que o conector tem de cumprir para dispor das funcionalidades básicas.

⁸<https://github.com/International-Data-Spaces-Association/Java-Representation-of-IDS-Information-Model>

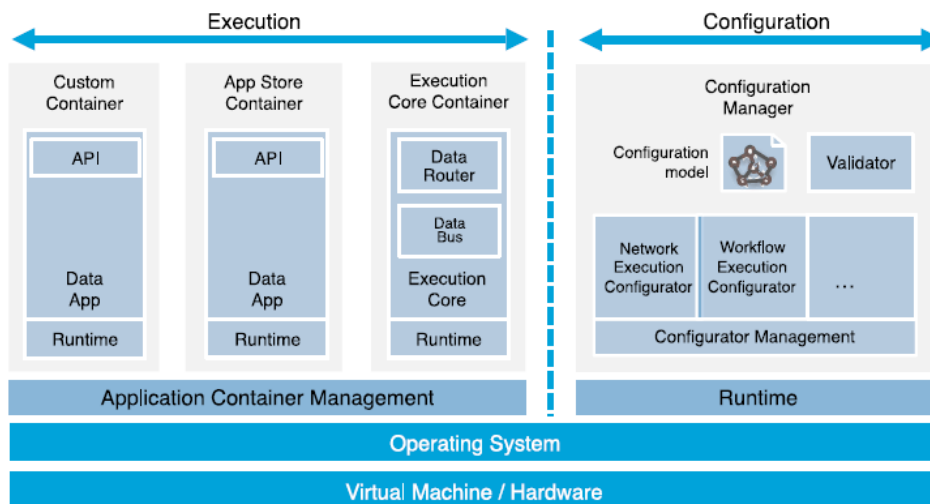


Figura 8: Arquitetura de referência de um IDS Connector [31].

Segundo a arquitetura de referência de um IDS Connector, o conector é formado por duas fases, a Fase de Execução (Execution) e a Fase de Configuração (Configuration).

A **Fase de Execução (Execution)** tem o objetivo de disponibilizar os dados, para os restantes conectores. Para tal, são definidos três tipos de containers, que devem estar presentes na arquitetura, o Execution Core Container, o App Store Container e o Custom Container. De forma a existir um isolamento entre as Data Apps, é atribuído um Custom Container a cada uma. As Data Apps, são responsáveis por processar e disponibilizar os dados, o número de Data Apps vai depender das necessidades da organização que está a desenvolver o conector. O Execution Core Container, possui as componentes indispensáveis no desenvolvimento das interfaces de comunicação para as Data Apps. O App Store Container, é um container certificado pela IDSA, e é obtido através de um dos componentes do IDS, a App Store. A gestão destes containers é feita por um Application Container Management [31].

Na **Fase de Configuração (Configuration)**, encontram-se as configurações do conector que são especificadas através do Configuration Model. O Configuration Model, descreve a configuração de um conector, é validado pelo Validator presente no Configuration Manager, e é independente da tecnologia utilizada no desenvolvimento do conector através da utilização dos Execution Configurators. É possível criar várias configurações para diferentes fases do conector (desenvolvimento, teste e produção) [31].

O Usage Control, é uma extensão ao atual típico controlo de acesso que permite às empresas definirem regras para processamento dos seus dados. O IDS suporta data-centric Usage Control ao adicionar políticas (IDS Policies) nos dados que estão a ser partilhados, de modo

que os Data Owners consigam ter controlo total dos dados durante todo o seu percurso [34]. As políticas adicionadas aos dados, são definidas pelos seus titulares através de Contracts, que possuem um conjunto de restrições. O IDS Connector é responsável por aplicar essas políticas.

Anteriormente, foram apontadas as entidades conceptuais, que podemos encontrar dentro do IDS. Contudo neste trabalho, o foco está na comunicação entre os Consumers (Data Consumers) e os Providers (Data Providers). O Provider tem a responsabilidade de fornecer os dados e de configurar o respetivo IDS Connector, de forma a disponibilizar os dados aos restantes IDS Connectors, presentes no IDS [31]. O papel do Consumer é, através do próprio IDS Connector, estabelecer uma ligação segura com o conector do Provider, de maneira a receber os dados provenientes das fontes conectadas ao Provider e autorizadas pelos seus titulares [31].

A partilha de dados entre Provider e Consumer, é efetuada através da interação entre os seus IDS Connectors. A interação entre esses conectores, conta com várias tarefas que precisam de ser cumpridas, de modo a finalizar a configuração de ambos os IDS Connectors para dar início à partilha de dados.

Todos os dados partilhados entre os IDS Connectors e as suas regras de utilização, devem ser representadas através de um conjunto de entidades, que estão presentes nos respetivos conectores. A seguinte imagem representa a relação entre essas entidades.

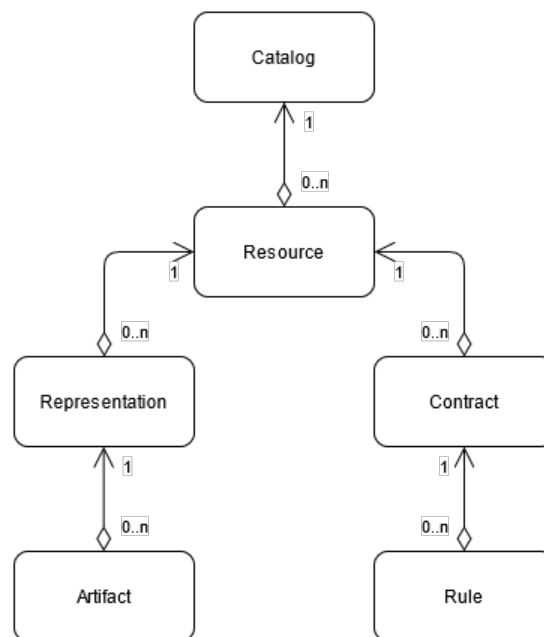


Figura 9: Relações entre as entidades do IDS Connector.

A partir da imagem apresentada é possível observar o modelo de dados do IDS Connector. O modelo de dados do IDS Connector, está estruturado com base no Modelo de Informação do IDS, e é composto por diversas entidades. A entidade principal é o Resource, que é organizado no Catalog, e pode conter várias Representations e Contracts. Uma Representation, descreve mais detalhadamente os dados disponibilizados aos Consumers, e inclui uma lista de Artifacts. O Artifact, disponibiliza um ponto de acesso direto aos dados, e apresenta as propriedades referentes a esses dados. Por fim, um Contract é atribuído a um Resource e possui um conjunto de Rules (Restrições). Como referido anteriormente, os Contracts são utilizados para definir as políticas (IDS Policies).

A comunicação entre conectores é feita recorrendo ao IDS Protocol, que conta com um conjunto de mensagens, as IDS Messages, que estão organizadas em várias classes e subclasses.

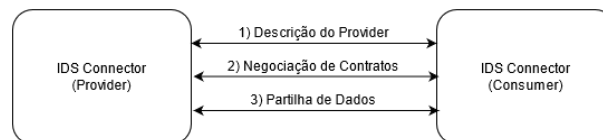


Figura 10: Principais interações entre os IDS Connectors.

Segundo a imagem exibida, podemos agrupar as classes e subclasses das IDS Messages em três interações:

1. **Descrição do Provider:** A Descrição do Provider, incorpora todas as mensagens usadas entre os conectores para enviar a descrição do Provider e das suas entidades, com o objetivo de comunicar ao Consumer, quais os dados e as respetivas condições para iniciar a partilha;
2. **Negociação de Contratos:** A Negociação de Contratos, corresponde às mensagens utilizadas para estabelecer os acordos entre os conectores, e definir as regras de acesso e controlo dos Resources e dados partilhados;
3. **Partilha de Dados:** A Partilha de Dados, agrupa todas as mensagens usadas na troca de dados e no envio das notificações ao Consumer, quando ocorrem atualizações nos Resources do Provider.

3.2.2 IDS-RAM vs RAMI 4.0

A quarta revolução industrial e a introdução do conceito de Fábricas Inteligentes originou novos problemas tecnológicos e organizacionais, que obrigaram as empresas a apostarem em modelos capazes de os solucionar, como o RAMI 4.0 e o IDS-RAM. O RAMI 4.0, representa um modelo mais conceptual e o seu foco está na integração dos vários ativos industriais, presentes no chão de fábrica com os serviços e aplicações presentes em níveis superiores na organização [30]. O IDS-RAM, consiste num modelo mais adaptado à realidade focando-se em determinados aspetos do modelo RAMI 4.0 ao oferecer soluções específicas para esses aspetos, de forma a assegurar a confiança e a segurança na partilha de dados interorganizacional, mantendo a governação dos dados [30].

Apesar de cada modelo atacar diferentes desafios na Indústria 4.0, ambos os modelos partilham a preocupação com a interoperabilidade entre sistemas [30]. Os dois modelos possuem a Camada de Negócio (Business Layer) e a Camada Funcional (Functional Layer), mas apesar de possuírem o mesmo nome, o seu significado é diferente. No RAMI 4.0 a Camada de Negócio representa o modelo de negócio e estabelece a ligação entre os diversos processos de negócio, no IDS-RAM a Camada de Negócio é responsável por definir e categorizar os papéis dos participantes no IDS. Relativamente à Camada Funcional, o RAMI 4.0 descreve formalmente todas as funções de um componente pertencente à Indústria 4.0 e o IDS-RAM, descreve os requisitos funcionais do IDS.

As maiores diferenças entre estes dois modelos encontram-se nas perspetivas de Segurança, Certificação e Governação, encontradas no IDS-RAM, e nos eixos horizontais da esquerda "Ciclo de Vida e Fluxo de Valor" e da direita "Níveis de Hierarquia" presentes no RAMI 4.0 [30]. As perspetivas do IDS-RAM, representam os principais valores que um IDS oferece. O "Ciclo de Vida e Fluxo de Valor" e os "Níveis de Hierarquia" do RAMI 4.0, descrevem o ciclo de vida do produto e todos os níveis hierárquicos dentro da organização.

O objetivo do trabalho, sublinha os problemas relacionados com a partilha de informação e da interoperabilidade num contexto interorganizacional, logo é importante estudar o modelo que mais se adapta a esses problemas. O IDS-RAM, é o modelo que mais se adequa para resolver esse tipo de problemas, e por essa razão será o modelo abordado neste trabalho.

3.2.3 Implementações

Atualmente, várias organizações e instituições de investigação implementaram as suas próprias versões do IDS-RAM e do IDS Connector. Neste ponto são apresentadas as principais organizações que estão a apostar na implementação do IDS-RAM e de IDS Connectors. Além disso, também são apresentados futuros projetos onde se utilizam IDS Connectors.

3.2.3.1 Fraunhofer

A Fraunhofer é uma organização privada sem fins lucrativos, dedicada à investigação e ao desenvolvimento de novas tecnologias e conceitos capazes de gerar valor para os seus clientes e parceiros, sendo uma delas os International Data Spaces. A Fraunhofer comanda a Strategy Initiative Data Spaces, como um programa de investigação dedicado ao desenvolvimento contínuo dos princípios do IDS-RAM [31].

Através do IDS Lab, a Fraunhofer oferece suporte e documentação para ajudar na compreensão dos International Data Spaces. Dentro do IDS Lab é possível encontrar várias implementações dos perfis de IDS Connectors, para serem usados em cenários específicos. Durante este trabalho, foram estudados dois tipos de implementações de IDS Connectors, o Trusted Connector e o DataSpace Connector.

O **Trusted Connector**, dá ênfase na segurança, ao possuir mecanismos como Secure Boot, Remote Platform Attestation e Data Usage Control através da LUCON Framework ⁹, que aumentam o nível de segurança na partilha de dados entre os participantes do IDS em relação às restantes versões de IDS Connectors. O Trusted Connector, possui quatro funcionalidades principais [35]:

1. **Isolamento de Apps (App Isolation)**: As aplicações dentro do Trusted Connector estão isoladas entre si para evitar perdas de dados e falhas nas restantes aplicações [35];
2. **Gestão de Identidade Interorganizacional (Cross-Enterprise Identity Management)**: Os conectores que estão ligados ao IDS possuem um identificador único dentro de um certificado X.509 para os identificar [35];

⁹<https://www.dataspaces.fraunhofer.de/en/software/usage-control/lucon.html>

3. **Plataformas de Confiança (Trusted Platform):** O Trusted Connector utiliza o trust-me¹⁰, em alternativa ao Docker, como uma camada segura para a gestão dos containers [35];
4. **Controlo na Utilização dos Dados (Data Usage Control):** O controlo do fluxo de dados é feito através de políticas criadas a partir da LUCON Framework [35].

O **DataSpace Connector**, consiste num projeto "open-source" no qual estão a contribuir para o seu desenvolvimento diversas organizações e institutos de investigação, e tem o objetivo de oferecer uma forma simples e fácil das organizações entrarem no ecossistema IDS, ao adicionar as funcionalidades básicas de um IDS Connector a sistemas já existentes [36]. Além disso, o DataSpace Connector serve como base para o desenvolvimento de novas componentes para serem integradas no ecossistema IDS. O DataSpace Connector utiliza a versão do Modelo de Informação do IDS mais recente e um serviço de mensagens, o IDS Messaging Service para comunicar com os restantes componentes do IDS. Dentro do IDS é também possível encontrar uma REST API para fazer a gestão dos seus recursos. Por fim, o DataSpace Connector possui um conjunto de funcionalidades que o ajudam a ser um conector fácil de integrar, e capaz de isolar o Modelo de Informação do IDS dos sistemas conectados [36].

Dos conectores estudados, foi escolhido o DataSpace Connector pois para além de simplificar o processo de integração de software com o ecossistema IDS, também é dos conectores mais atualizado e que mais se enquadra neste trabalho.

3.2.3.2 FIWARE

Em qualquer solução inteligente existe a necessidade de obter, processar e disponibilizar os dados a várias entidades, de forma a adicionar valor às decisões tomadas por essas entidades, através da informação de um determinado contexto. A iniciativa FIWARE, definiu um conjunto de padrões universais para a gestão de dados, com o objetivo de acabar com essa necessidade.

A FIWARE, corresponde a um ambiente de desenvolvimento onde estão presentes várias componentes de uma plataforma "Open Source", que podem ser integradas entre si ou com componentes de terceiros, com o intuito de construir plataformas que suportam o desenvolvimento

¹⁰<https://trustm3.github.io/>

de soluções inteligentes. O núcleo de um ecossistema FIWARE é a plataforma FIWARE, que oferece um conjunto de APIs que facilitam o desenvolvimento rápido de soluções inteligentes em contexto IoT. A FIWARE disponibiliza uma implementação de referência das componentes FIWARE, com o objetivo de ajudar as organizações a apresentar propostas com um menor custo possível [37].

A componente obrigatória para o desenvolvimento de soluções inteligentes baseadas em FIWARE, é o FIWARE Context Broker. O FIWARE Context Broker, tem o objetivo de fazer a gestão de todo o ciclo de vida da informação contextualizada e fornecer acesso às restantes componentes. De um ponto de vista mais tecnológico, o FIWARE Context Broker, possui uma implementação de referência, o FIWARE-Orion Context Broker. Esta implementação de referência conta com uma REST API baseada na Open Mobile Alliance's Next Generation Service Interface (NGSI) e uma base de dados orientada a documentos [38].

De forma a facilitar o desenvolvimento das soluções inteligentes, a FIWARE conta com vários FIWARE Generic Enablers (FIWARE GE), que têm de estar conectados ao FIWARE Context Broker, e o seu objetivo passa por cumprir com os requisitos mais frequentes nos vários tipos de soluções inteligentes. Os FIWARE GE, oferecem uma ou mais componentes, que podem cooperar entre si ou com componentes de terceiros. É possível organizar os FIWARE GE da seguinte forma:

- Integrações com dispositivos e/ou componentes, para obter os dados e atualizações;
- Gestão de dados, com o objetivo de aplicar regras de controlo de uso e de acesso aos dados;
- Processar, analisar e visualizar os dados, de forma a assistir os utilizadores finais na tomada de decisões.

Um dos objetivos da FIWARE, é criar um mercado de soluções e plataformas FIWARE desenvolvidas pela comunidade, no sentido de expandir o ecossistema FIWARE e facilitar, acelerar e reduzir o custo do desenvolvimento das soluções inteligentes [39].

Os IDS Connectors, possibilitam a partilha de dados de forma segura entre participantes do IDS, sem comprometer com a sua integridade, e permitem aos Data Owners definirem regras para controlarem o uso e acesso dos seus dados. Os conectores, assim como o FIWARE Context Broker, proporcionam formas de fazer integrações com componentes de terceiros, de

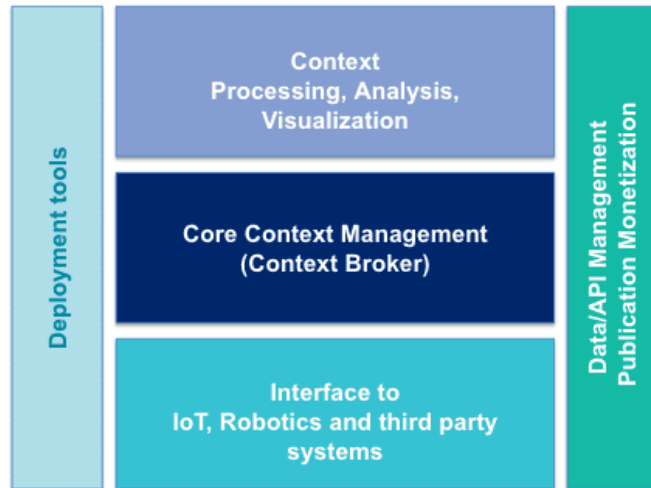


Figura 11: Context Broker [37].

forma a acrescentar novas funcionalidades. Ao integrar o FIWARE Context Broker com um IDS Connector, torna-se possível fazer uma integração com os FIWARE Generic Enablers, com o objetivo de adicionar um conjunto de funcionalidades, pertencentes à plataforma FIWARE, nos IDS Connectors.

A seguinte imagem representa um exemplo de uma possível integração de um IDS Connector com um FIWARE IoT GE, para facilitar a obtenção dos dados proveniente de sensores e/ou dispositivos.

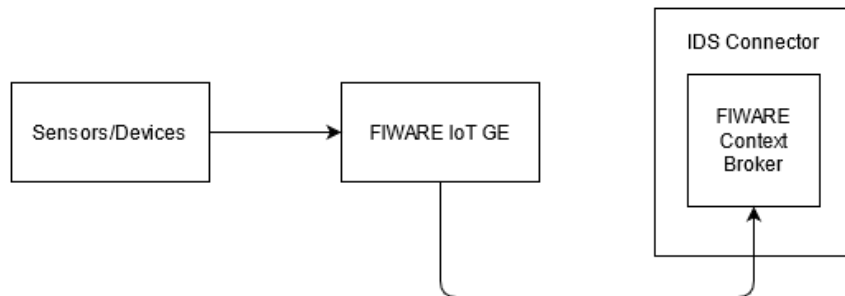


Figura 12: Exemplo de integração de um FIWARE GE com um IDS Connector.

3.2.3.3 Outras Implementações

As duas organizações apresentadas anteriormente correspondem às principais organizações que estão a apostar na implementação dos conceitos do IDS. Contudo, existem outras organizações, que começaram a desenvolver as suas próprias versões do IDS Connector. É importante referir algumas dessas organizações.

A Engineering Ingegneria Informatica ¹¹ desenvolveu o TRUE (TRUsted Engineering) Connector, com o objetivo de facilitar a partilha de dados mantendo a sua governação, ao implementar conceitos do projeto GAIA-X. O conector está presente no catálogo FIWARE e é uma das principais componentes para criar espaços de dados baseados em FIWARE.

A Eclipse Foundation ¹² propôs um novo tipo de IDS Connector, o Eclipse Dataspace Connector, que foi aprovado e encontra-se na fase de desenvolvimento. O conector tem o objetivo de implementar os conceitos do International Data Space e do projeto GAIA-X, de forma a garantir a governação e a partilha interorganizacional dos dados.

Depois de realizada a análise dos IDS Connectors, foi encontrada uma falha na implementação do paradigma "Publish and Subscribe", onde a sua implementação não se encontrava de acordo com o padrão típico do paradigma. Uma das alternativas encontradas, foi a simulação do paradigma onde um determinado IDS Connector realizava pedidos de dados regularmente a outro IDS Connector, de forma a verificar se existiam possíveis atualizações nos dados. Contudo, no final do desenvolvimento deste trabalho, verificou-se alguns avanços na implementação deste tipo de funcionalidades.

3.3 GAIA-X

Os dados possuem um enorme valor para as empresas, e por isso as empresas procuram formas de os partilhar. Surgem novas questões e conceitos relacionados com a privacidade e a segurança dos dados partilhados, como a Governação dos Dados (Data Sovereignty). A Governação dos Dados, consiste no controlo e governação total sobre os dados, por parte dos seus titulares independentemente da sua localização [40]. A arquitetura do projeto GAIA-X define soluções técnicas para estabelecer a Governação dos Dados de acordo com os standards da União Europeia [40].

O projeto GAIA-X, consiste numa nova iniciativa, que segue a mesma visão da iniciativa IDS de promover a governação dos dados e de criar um ecossistema onde a partilha de dados é realizada de forma segura e confiável, ao possibilitar a interoperabilidade e a portabilidade das infraestruturas de dados e serviços, de acordo com a European Data Strategy¹³ [41]. Assim

¹¹<https://www.eng.it/en/case-studies/true-connector-per-facilitare-la-condivisione-di-dati-in-gaiax>

¹²<https://projects.eclipse.org/proposals/eclipse-dataspace-connector>

¹³<https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/european-data-strategy>

sendo, a iniciativa GAIA-X propôs uma arquitetura, onde une novos conceitos, relacionados com a Governança dos Dados, com alguns dos conceitos presentes no IDS-RAM [41].

A arquitetura do GAIA-X, aponta para uma rede descentralizada, assim como o IDS-RAM, onde múltiplos sistemas seguem um standard comum [41]. Para que a arquitetura consiga cumprir com a visão do GAIA-X, deve estar de acordo com os princípios base da arquitetura, que são: a Abertura e Transparência (Openness and Transparency), a Interoperabilidade (Interoperability), os Sistemas de Federação (Federated Systems) e a Autenticidade e Confiança (Authenticity and Trust) [41].

A iniciativa GAIA-X, disponibiliza um conjunto de normas técnicas, que a arquitetura deve seguir para impor os princípios apresentados anteriormente e assegurar a sua visão.

1. **Segurança desde o Desenho (Security-by-Design):** O GAIA-X, coloca a segurança tecnológica como um dos alicerces da arquitetura, para proteger todos os participantes e sistemas presentes no ecossistema do GAIA-X [40];
2. **Privacidade desde o Desenho (Privacy-by-Design):** A União Europeia é exigente no que toca aos regulamentos sobre a privacidade, e por isso a arquitetura já considera todos os aspetos relacionados com a mesma [40];
3. **Permitir a Federação, Distribuição e Descentralização (Enabling Federation, Distribution e Decentralisation):** A arquitetura deve tomar em consideração abordagens como a distribuição, federação e a descentralização [40];
4. **Simplicidade e Facilidade no Uso (Usage-Friendliness and Simplicity):** Uma boa experiência de utilização, o uso de standards, protocolos abertos, e processos simples são cruciais para a adoção do GAIA-X [40];
5. **Processamento em Máquinas (Machine-Processability):** Tudo o que é partilhado dentro do GAIA-X deve ser "machine-readable". Os sistemas devem expor APIs como meio principal de interação no GAIA-X. O uso de interfaces para as APIs, aumenta a interação das pessoas com o GAIA-X, sendo que a automação é suportada na arquitetura [40];
6. **Representação Semântica (Semantic Representation):** A arquitetura deve assegurar que o modelo de dados do GAIA-X é estabelecido, para garantir a interoperabilidade entre os sistemas [40].

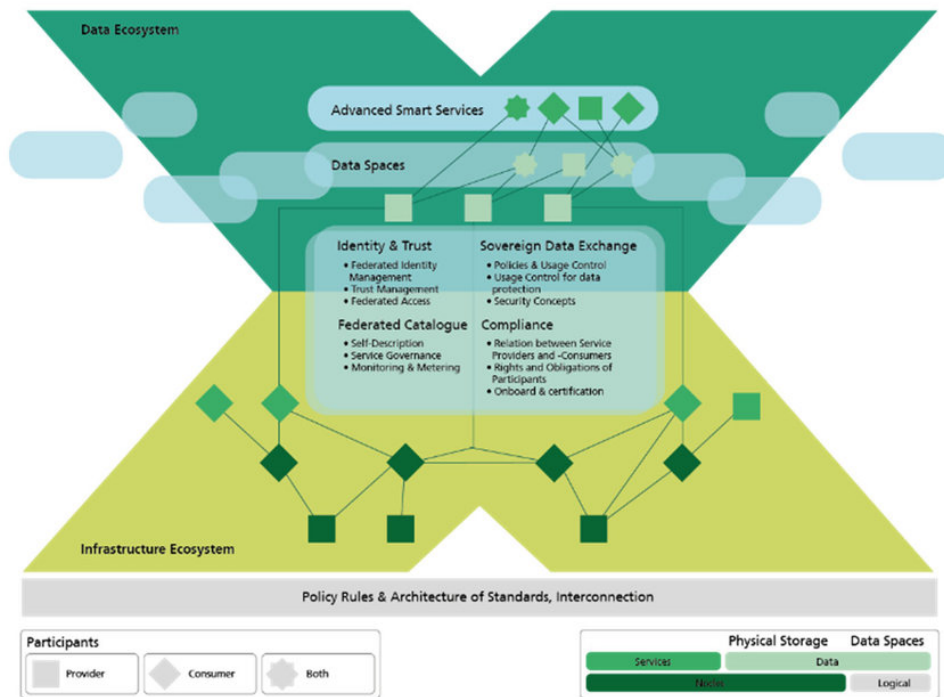


Figura 13: Arquitetura do GAIA-X [41].

A partir da imagem apresentada na figura 13 podemos observar, que o ecossistema GAIA-X é formado por um Ecosistema de Infraestrutura (Infrastructure Ecosystem) e um Ecosistema de Dados (Data Ecosystem), ambos conectados através de vários Serviços de Federação (Federation Services). O Ecosistema de Infraestrutura, é composto por todos os serviços, representados através do ativo *Node*, e as componentes necessárias para armazenar, processar e partilhar dados presentes no GAIA-X [40]. O Ecosistema de Dados, define os dados como o principal ativo de negócio, ao representá-los através do ativo *Data* [40]. Semelhante ao IDS, o GAIA-X classifica os seus participantes como Fornecedores (Providers) e Consumidores (Consumers) de Dados. Também é possível observar, na imagem, que na base da arquitetura estão presentes as Regras de Políticas e uma Arquitetura de Standards (Policy Rules & Architecture of Standards, Interconnection).

Como foi referido no paragrafo anterior, os Serviços de Federação são utilizados para estabelecer a conexão entre o Ecosistema de Infraestrutura e o Ecosistema de Dados, e por isso são considerados como o núcleo do GAIA-X. Estão presentes nos Serviços de Federação, os serviços de Identidade e Confiança (Identity and Trust), os serviços de Conformidade (Compliance), os Catálogos Federados (Federated Catalogues) e os serviços de partilha para Dados Governados (Sovereign Data Exchange) [41].

O GAIA-X e o IDS partilham soluções e conceitos relacionados com a partilha de informação

numa rede descentralizada onde a segurança e a confiança são cruciais para o seu funcionamento. Contudo, enquanto o IDS dedica-se a resolver as questões relacionadas com a partilha de dados interorganizacional, o GAIA-X, foca-se na forma como esses dados são armazenados, processados e partilhados, ou seja, o seu principal objetivo é assegurar a Governança e infraestrutura dos Dados. Como o foco deste trabalho são os problemas relacionados com a partilha de informação, torna-se irrelevante seguir a abordagem GAIA-X.

3.4 Desafios para as PME

A partilha de informação interempresarial conta com um conjunto de problemas relacionados com a segurança, confiança, compatibilidade e a flexibilidade, aos quais as pequenas e médias empresas têm vindo a tentar solucionar através da adoção das abordagens apresentadas no capítulo anterior.

Uma das abordagens apresentadas anteriormente, o International Data Space, pretende solucionar grande parte dos problemas, relacionados com a partilha de informação interempresarial. Contudo, a abordagem orientada a IDS não é perfeita, e por isso a sua adoção apresenta um conjunto de barreiras e desafios, que requerem um grande investimento, impensável para as pequenas e médias empresas. Os principais desafios e barreiras são:

- **Complexidade na implementação:** Um dos objetivos do IDS, passa por criar uma rede entre todas as entidades, que abrange todo o ecossistema do negócio. A criação desta rede, obriga às empresas a realizarem grandes investimentos, dos quais pequenas e médias empresas não possuem a capacidade de os fazer. Uma das soluções impostas pela IDSA para solucionar este problema passa por criar infraestruturas regionais e sectoriais especializadas e capacitadas tecnologicamente, permitindo às empresas descarregarem da *cloud* os conectores necessários e apropriados para usufruírem dos IDSs existentes.
- **Gestão de dados:** Um dos aspetos fundamentais do IDS é a partilha de dados entre os seus participantes. Uma vez que, muitos dos dados partilhados por todo o IDS são considerados dados em bruto, é necessário um grande esforço, por parte das empresas, para o seu processamento.
- **Implementação dos padrões impostos pelo IDS:** De forma a assegurar a interoperabilidade

bilidade e a compatibilidade entre sistemas, o IDS, conta com um conjunto de padrões, definidos no Information Model pela IDSA. A implementação desses padrões, requerem grandes investimentos, que dificultam a adoção do IDS por pequenas e médias empresas.

- **Certificação:** A fim de criar uma "trusted-network" entre os seus participantes, com o intuito de possibilitar a partilha de dados de forma segura, o IDS possui um processo de certificação bastante complexo. A certificação consiste na submissão de todos os participantes e componentes presentes no IDS a uma avaliação, segundo um conjunto de critérios com o objetivo de validar e determinar o seu grau de confiança perante todo o IDS.
- **Falta de um Base Connector:** O IDS Connector, usado por todos os participantes, corresponde à componente principal do IDS. O IDS abrange vários tipos de empresas, cada empresa possui um conjunto de requisitos específicos e únicos. Como o IDS não apresenta um Base Connector, que funcione em todas as empresas, as empresas vêm-se confrontadas com a necessidade de desenvolver o seu próprio IDS Connector, a partir da arquitetura de referência, para poder participar no IDS. A criação de novos conectores, é um processo bastante complexo, que impossibilita as pequenas e médias empresas aderir ao IDS.
- **Especialização:** O IDS, apresenta uma arquitetura complexa e um conjunto de padrões e critérios que devem ser cumpridos. Isto obriga as empresas a contratar alguém especializado, para a sua implementação e configuração.

Vimos que a partilha de informação interorganizacional conta com diversos desafios que devemos ter em atenção, como a flexibilidade, a compatibilidade, a segurança e confiança entre sistemas e entidades. Com base no estudo dos modelos apresentados anteriormente, o IDS-RAM é o que mais se adequa para responder a esses desafios. Assim sendo, os International Data Spaces foram escolhidos para o desenvolvimento de uma arquitetura que permita a partilha de informação interorganizacional.

O presente trabalho não pretende solucionar de todo os desafios apresentados, mas sim contribuir para a base de conhecimento existente, com casos de uso adicionais, devidamente contextualizados, para apoiar as pequenas e médias empresas na sua jornada de transformação digital.

Capítulo 4

Arquitetura IDS para a partilha de informação interorganizacional

A análise do estado da arte evidencia os IDS como uma abordagem pragmática no suporte à partilha de informação 4.0 no contexto interorganizacional. No entanto, enquanto modelo arquitetural emergente, pretende-se estudar a exequibilidade prática deste novo paradigma, no contexto da troca de informação interorganizacional em tempo real. Deste modo, no presente capítulo discute-se uma arquitetura IDS, com base nos seguintes pressupostos:

- A arquitetura IDS, deve suportar um modelo de "Push Data", ou seja, o cliente deve estar "constantemente" a receber dados sem a necessidade de enviar pedidos previamente. Esta necessidade assíncrona é fundamental num contexto de troca de dados em tempo real. O paradigma propõe que a entidade responsável pelo envio de dados, categorize e distribua os dados em diferentes tópicos, de modo que outras entidades possam estar à escuta desses tópicos para obter os dados. O envio dos dados é despoletado através de eventos (ex.: alteração no estado de uma máquina) determinados pela entidade emissora. Este modelo promove a escalabilidade, permite fluxos de trabalho assíncronos numa organização, reduz o tempo de resposta e permite uma integração mais simples entre sistemas.
- A arquitetura IDS, deve assegurar a compatibilidade com os diversos sistemas e protocolos. O universo tecnológico de cada organização é muito heterógeno o que torna difícil assegurar a compatibilidade entre vários sistemas. A arquitetura IDS contempla

a uniformização da troca de dados entre organizações. No entanto é necessário assegurar mecanismos para a troca de informação entre IDS e os Sistemas de Informação Organizacionais das empresas.

- A partilha de dados, na arquitetura IDS, deve ser realizada de forma segura e sem comprometer com a integridade dos dados. Devido à valorização dos dados, cada vez mais a segurança se torna indispensável para partilha de dados interorganizacional.

O trabalho de especificação e experimentação da arquitetura IDS, obedece a uma metodologia de investigação cujo modelo se descreve na secção seguinte.

4.1 Metodologia

A metodologia de base para o desenvolvimento deste trabalho e em particular o desenvolvimento da arquitetura IDS, seguiu um modelo incremental e iterativo, ou seja, o trabalho contemplou diversas fases, com uma ou mais interações cada uma. Em cada fase, são aproveitados os resultados obtidos em fases anteriores, e são realizadas revisões para o aperfeiçoamento de componentes já desenvolvidas. Sejam estes componentes artefactos mais teóricos ou práticos. A figura 14, apresenta o plano de trabalho utilizado no desenvolvimento da dissertação.

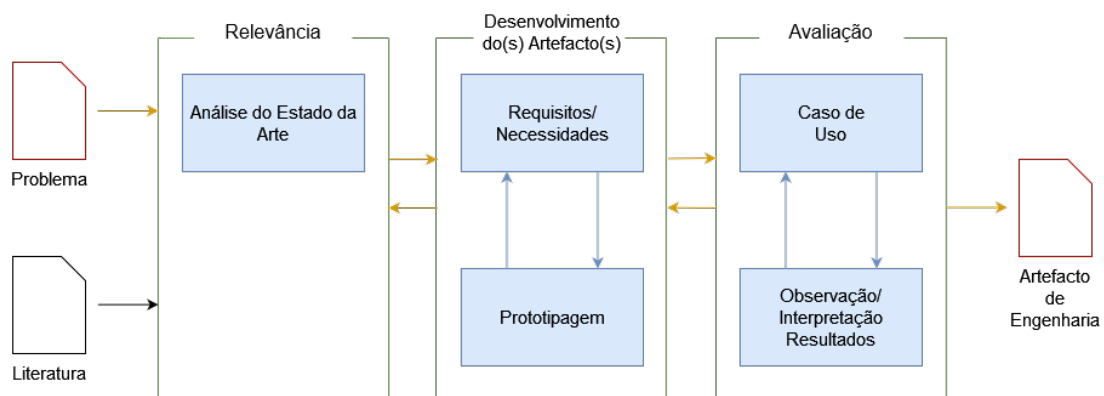


Figura 14: Plano de Trabalho.

A partir da imagem apresentada na figura 14, é possível identificar as três fases que compõem o plano de trabalho utilizado para a realização da dissertação:

- **Relevância:** Identificado o possível Problema, a fase da Relevância, tem o objetivo de validar a relevância do Problema com base na Análise do Estado da Arte. De modo

a cumprir com esse objetivo, foram utilizadas várias bases de dados científicas como o Google Scholar, o ResearchGate e o IEEEExplore, onde se realizaram várias pesquisas específicas, no âmbito da partilha de informação no contexto da Indústria 4.0, sobre os diversos modelos, abordagens e soluções utilizadas na partilha de informação interorganizacional. Além disso, também foram estudadas soluções mais profissionais e artigos de organizações (White Papers), que estão atualmente a impulsionar os IDSs. Depois de construído o repositório de informação útil para a Análise do Estado da Arte, escolheram-se apenas os artigos que apresentaram modelos, abordagens ou soluções compatíveis com o RAMI 4.0. Os restantes artigos, como apresentam modelos, abordagens e soluções que fogem aos pressupostos do RAMI 4.0, tornam-se irrelevantes e por isso não foram abordados neste trabalho.

- **Desenvolvimento do(s) Artefacto(s):** Durante a fase do Desenvolvimento do(s) Artefacto(s), realizou-se o desenvolvimento da arquitetura IDS seguindo um conjunto genérico de necessidades previamente definidas, e na experimentação de diversas soluções, de modo a viabilizar a arquitetura especificada. Além disso, esta fase contou com a iteração entre a Prototipagem e os Requisitos/Necessidades, de forma a ser possível experimentar diferentes frameworks e tecnologias no desenvolvimento dos protótipos, o que permitiu refinar as necessidades definidas. No decorrer desta fase, existiu vários momentos onde foi necessário regressar à fase da Relevância, para obter mais informação e conseguir responder a determinadas questões que foram surgindo durante o desenvolvimento.
- **Avaliação:** Por fim, a fase da Avaliação consiste na avaliação qualitativa dos artefactos desenvolvidos na fase anterior perante um Caso de Uso, através da observação e avaliação da sua utilidade e exequibilidade para a partilha de informação num contexto interorganizacional. É de notar que nesta fase também foi possível regressar à fase de Desenvolvimento do(s) Artefacto(s), para a fazer as alterações necessárias de forma a corrigir os problemas que foram surgindo durante a avaliação.

No final da metodologia, obteve-se o Artefacto de Engenharia, que consiste num protótipo da solução tecnológica, desenvolvido com base nos princípios do IDS, que respeita grande parte dos pressupostos do modelo IDS-RAM.

4.2 Protótipo IDS

O principal propósito deste trabalho passa por explorar e desenhar soluções baseadas no IDS-RAM, que permitam assegurar o fluxo de informação automatizado e o mais aproximado do "tempo real", entre o ambiente intra e interorganizacional. Esta automatização requer negociar e acordar aspetos relacionados com o conteúdo e a validade temporal do conteúdo a partilhar, sobre a garantia de uma infraestrutura segura e que garanta a propriedade dos dados. O uso dos IDSs no contexto de partilha de informação entre organizações levanta vários problemas e restrições tais como:

- De forma a validar a solução, segundo a IDSA, é obrigatório utilizar as mensagens definidas no modelo IDS para estabelecer a comunicação entre os conectores IDS. Cada mensagem possui um objetivo específico e deve ser usada exatamente no contexto para o qual foi definida. Um exemplo disso é a Notification Message, que representa um tipo de mensagens que só podem ser utilizadas para efeitos informativos, no qual não é esperada nenhuma resposta por parte do recetor. Nenhuma outra forma de informação/notificação básica deve ser usada na comunicação entre conectores;
- Não é permitido alterar a integridade dos conectores IDS, o que nos obriga a desenvolver componentes "Plug and Play", ou seja, componentes que podem ser integradas facilmente com o conector, de forma a adicionar novas funcionalidades ao mesmo;
- Um conector IDS não pode identificar nem comunicar diretamente com as DataApps presentes nos restantes conectores do IDS, pois as DataApps estão presentes nas redes internas das empresas, isoladas do mundo exterior, e segundo os princípios do IDS só é permitido aceder a essas redes através dos conectores IDS.

No decorrer do estudo dos IDSs, foram criados vários protótipos, não só para compreender o modelo IDS, como também para avaliar o funcionamento dos conectores. Para além disso o protótipo final deverá considerar um conjunto de restrições e necessidades/requisitos discutidos anteriormente, e estar dentro do quadro de objetivos definidos na introdução. O que se discute de seguida foca apenas no protótipo final e não na descrição da evolução até chegar ao mesmo.

4.2.1 Integração com Apache Camel

O uso de Apache Camel no contexto da especificação da arquitetura IDS, tem um duplo propósito:

1. Os paradigmas clássicos para a troca de informação assentam em mecanismos síncronos de Pedido/Resposta com mensagens tipificadas. Os IDSs pretendem que essas mensagens possam incluir a comunicação do estado dos ativos do chão de fábrica em tempo real, pelo que o uso do paradigma de comunicação "Publish and Subscribe" é fundamental. Adicionalmente, os conectores IDS estudados não implementam "Publish and Subscribe", apenas o simulam por meio de pedidos que ocorrem em intervalos de tempo previamente especificados. O Apache Camel ajuda na implementação efetiva do paradigma. A implementação do paradigma "Publish and Subscribe", permite cumprir com o propósito da arquitetura IDS de suportar um modelo de "Push Data". Essa integração possibilita ao conector IDS, identificar possíveis atualizações nas fontes de dados, especificadas nos seus **Resources**, e conseqüentemente enviar "IDS Messages" de notificação, automaticamente para os restantes conectores.
2. Como a IDS é uma rede "pay-as-you-go", um dos seus objetivos é expandir constantemente o número de sistemas presentes, o que obriga os conectores IDS a serem compatíveis com vários sistemas, que utilizam diversos protocolos de comunicação e integração. Ao incorporar o Apache Camel ¹ com o conector IDS, é possível integrar o conector com as diferentes aplicações empresariais e os vários protocolos de comunicação existentes na organização, através de Enterprise Integration Patterns (EIP) [42]. A possibilidade da interligação entre os conectores IDS e um serviço baseado em Apache Camel, aumenta a compatibilidade e interoperabilidade de soluções baseadas em IDS, sem abdicarmos dos princípios de desenvolvimento baseado em normas e standards existentes. O Apache Camel, através de uma DSL (Domain Specific Language), nomeadamente a Apache Camel DSL, permite de uma forma mais ágil a definição de regras e rotas de integração, baseadas em um ou mais EIP para a partilha de dados. Além disso, o Apache Camel consegue operar com vários protocolos de comunicação e estruturas de dados, acessíveis através da utilização de URIs. A componente desenvolvida neste trabalho utiliza o Java DSL ², com o objetivo de criar as Rotas HTTP, ao estender a classe

¹<https://camel.apache.org/>

²<https://camel.apache.org/manual/latest/java-dsl.html>

do **RouteBuilder** ³.

Neste protótipo, os conectores IDS utilizam um serviço baseado na framework Camel, permitindo assim definir uma arquitetura IDS que cumpre com as restrições e os critérios necessários para poder ser aplicado num caso de uso. Podemos definir o serviço baseado em Camel, como uma componente "Plug and Play", que comunica com o conector IDS e restantes aplicações por meio de um conjunto de rotas, que são definidas através de uma REST API. A seguinte imagem (Figura 15) representa o Serviço Camel, conectado a um MQTT Broker e a um conector IDS, e cujo objetivo é estar à escuta das mensagens enviadas pelo MQTT Broker e enviá-las para o conector IDS, sempre que este as requisitar.

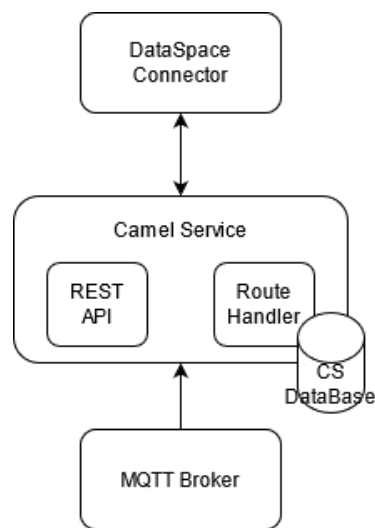


Figura 15: Visão genérica do Serviço Camel no contexto IDS.

A seguinte imagem (Figura 16) representa genericamente, a arquitetura do protótipo desenvolvido:

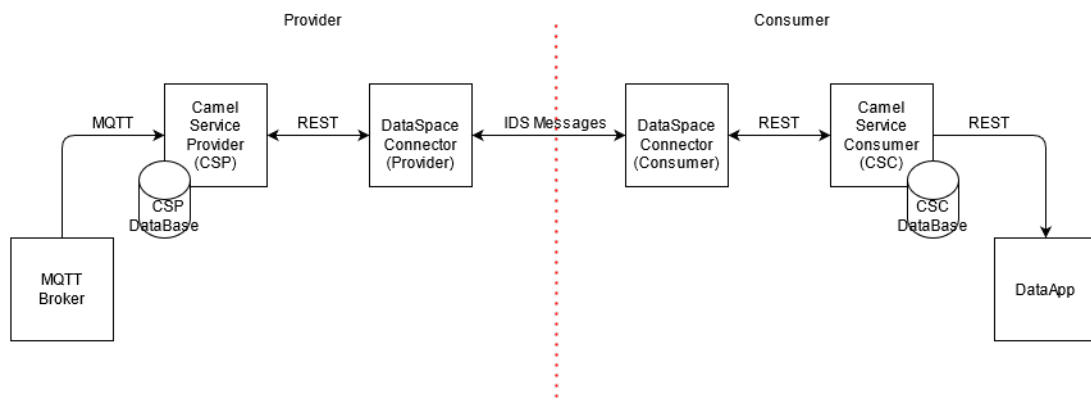


Figura 16: Visão geral da Arquitetura do Protótipo.

³<https://camel.apache.org/manual/latest/route-builder.html>

Podemos dividir a imagem apresentada na figura 16 em duas partes, do lado esquerdo estão todas as componentes relacionadas com a entidade que assume o papel de Provider e do lado direito com a entidade que assume o papel de Consumer. Em cada uma das implementações do conector IDS existe um Serviço Camel, o Camel Service Provider (CSP) e o Camel Service Consumer (CSC).

4.2.2 Detalhe da Arquitetura

A arquitetura do protótipo baseado em IDS, representado na figura 16, contém vários componentes, cujos papéis se escrevem de seguida. É de realçar que a presente arquitetura procura garantir:

- A comunicação baseada no IDS-RAM, seleccionando e implementando os conectores IDS mais adequados;
- Assegurar a interoperabilidade entre os conectores IDS e os sistemas internos das empresas;
- Usar protocolos e paradigmas de comunicação que permitam a troca de dados em tempo real e de forma assíncrona.

Não foi o foco desta arquitetura as questões relacionadas com a negociação de contratos, as regras de uso dos dados e a certificação dos participantes. Todas estas questões foram geradas através de configurações predefinidas.

4.2.2.1 Provider

O principal objetivo do Provider (lado esquerdo da figura 16), é receber e enviar as notificações provenientes do MQTT Broker para os respetivos Consumers, conectados via IDS. A fim de possibilitar essa interação, o Provider conta com um MQTT Broker, o CSP e seu próprio DataSpace Connector (Provider).

O MQTT Broker, surge aqui como um servidor para a troca de mensagens, segundo o modelo de "Publish and Subscribe". A utilização do protocolo MQTT, que pertence ao conjunto de

protocolos utilizados no âmbito da Indústria 4.0 ⁴, permite-nos garantir a troca de dados em tempo real e o assincronismo. Concretamente nesta arquitetura, o MQTT Broker disponibiliza os dados num tópico para que o Serviço Camel consiga estar à escuta e conseqüentemente obter os dados sempre que existir uma atualização nos mesmos.

O CSP é um serviço, cuja função é obter os dados dos tópicos presentes no MQTT Broker e disponibilizá-los para o DataSpace Connector (Provider), através de rotas criadas dinamicamente e em runtime. As rotas criadas, também permitem enviar pedidos ao DataSpace Connector (Provider), sempre que existir alguma atualização num dos tópicos, estabelecendo assim uma conexão entre os protocolos MQTT e REST, de modo a garantir o assincronismo e a partilha de dados em tempo real. Dentro do serviço CSP encontramos duas componentes, uma REST API e um Route Handler. A REST API disponibiliza um conjunto de pedidos para comunicar com o Route Handler, de modo a:

1. Criar, Apagar, Editar e Obter as rotas na base de dados do CSP;
2. Ativar e Desativar rotas específicas;
3. Obter determinados Payloads, nas diferentes rotas armazenadas na base de dados do CSP.

⁴<https://mqtt.org/>

O seguinte excerto de código, apresenta um exemplo de uma rota definida no CSP.

```
{
  "id": 1,
  "brokerURL": "tcp://mqtt-broker:1883",
  "brokerTopic": "broker-topic",
  "brokerPayload": "",
  "providerConnector": {
    "id": 1,
    "providerUrl": "http://provider:8081",
    "recipientUrl": "http://consumer:8080/api/ids/data",
    "resourceId": "http://provider:8081/api/offers/40c3ae49-c76e-
      4330-a4d8-6e50fe4d9aa7",
    "username": "admin",
    "password": "password"
  },
  "isActive": true
}
```

Exemplos de Código 1: Exemplo de uma Rota no CSP.

O DataSpace Connector (Provider), dedica-se a criar uma interface entre as fontes de dados e o IDS, neste caso o MQTT Broker por meio do CSP. Tudo o que está dentro da rede interna do Provider não tem conhecimento sobre as entidades, que estão dentro da rede interna do Consumer.

No Provider, é possível dividir o processo, que inicia a troca de dados em duas fases:

- Na **Primeira Fase**, é criada uma rota no CSP, que tem como objetivo notificar o DataSpace Connector (Provider) sempre que existir atualizações no MQTT Broker;
- Na **Segunda Fase**, o CSP recebe as atualizações do MQTT Broker e envia um conjunto de pedidos ao DataSpace Connector (Provider), que geram várias "IDS Messages", para notificar o DataSpace Connector (Consumer) de uma possível alteração num dos *Resources*.

4.2.2.2 Consumer

O objetivo do Consumer (lado direito da figura 16) é receber as notificações e os dados do DataSpace Connector (Provider), e redirecioná-los para as respectivas DataApps, através do CSC. O Consumer, é composto por uma DataApp, o CSC e o DataSpace Connector (Consumer).

A DataApp, representa a componente que recebe os dados do DataSpace Connector (Consumer), através do CSC. Um endpoint é disponibilizado pela DataApp, de modo a que o CSC consiga enviar os dados recebidos.

O CSC, representa um serviço, em que o seu principal objetivo é estabelecer a comunicação entre a DataApp e o DataSpace Connector (Consumer), através da criação de rotas. Sempre que existir uma atualização no lado do Provider, o DataSpace Connector (Consumer) recebe uma "IDS Message" de notificação, e utiliza o endpoint da rota criada no CSC para o notificar. A rota criada, depois de receber a notificação, executa um pedido ao DataSpace Connector (Consumer), que por sua vez comunica com o Provider, de modo a obter a última versão dos dados e posteriormente enviá-los para o endpoint da respetiva DataApp. Assim como no CSP, com a integração do CSC no conector IDS, conseguimos realizar a partilha de dados de forma assíncrona e em tempo real. Além disso, é possível garantir a interoperabilidade entre o conector IDS e os sistemas internos da empresa. Semelhante ao CSP, podemos encontrar uma REST API e um Route Handler no CSC, que adiciona as seguintes funcionalidades:

1. Criar, Apagar, Editar e Obter as rotas na base de dados do CSC;
2. Receber os dados e notificações do DataSpace Connector (Consumer);
3. Enviar os dados para as diferentes DataApps.

O seguinte excerto de código, apresenta um exemplo de uma rota definida no CSC.

```
{
  "id": 1,
  "dataAppURL": "http://data-app:1880/endpoint",
  "consumerConnector": {
    "id": 1,
    "consumerArtifactURI": "http://consumer:8080/api/artifacts/a9
      d76b8d-6d8d-4ec7-8d12-edf153bb5d17",
    "agreementURI": "http://provider:8081/api/agreements/84f7325c
      -1d62-4470-b03b-bb5603224ffb",
    "params": {},
    "username": "admin",
    "password": "password"
  }
}
```

Exemplos de Código 2: Exemplo de uma Rota no CSC.

O DataSpace Connector (Consumer), cria uma interface entre as DataApps e o IDS, em conjunto com o CSC. Assim como o DataSpace Connector (Provider), tudo o que está dentro da rede interna do Consumer não consegue identificar nem comunicar com as entidades, presentes na rede interna do Provider.

No Consumer, o processo responsável pelo envio dos dados às DataApps, é composto por três fases:

- Na **Primeira Fase**, é criada uma rota no CSC, com o propósito de receber as notificações do DataSpace Connector (Consumer) e enviar um pedido para obter os dados ao mesmo. A rota criada também possui o endpoint da DataApp, necessário para o envio dos dados.
- Na **Segunda Fase**, o DataSpace Connector (Consumer), recebe uma "IDS Message" do DataSpace Connector (Provider), a notificar uma possível alteração num dos seus *Resources*. Essa mensagem, irá acionar o envio do alerta para o CSC.
- Na **Terceira Fase**, após receber o alerta, o CSC envia um pedido ao DataSpace Connector (Consumer) para obter a última versão dos dados, e redirecioná-los para a respetiva DataApp, definida inicialmente na criação da rota.

4.2.3 Interações

Para melhor compreensão das interações entre os conectores IDS, considera-se o seguinte cenário: O Provider envia informação sobre o estado da máquina X, para o Consumer, que previamente subscreveu essa informação. A máquina não envia o estado diretamente para o Consumer, mas sim para um MQTT Broker. O Apache Camel Service atua como um mediador no sentido de garantir que a informação segue para o conector IDS no formato adequado em conformidade com os pressupostos do IDS-RAM.

O seguinte diagrama de sequência do protótipo desenvolvido (Figura 17), representa um conjunto de interações entre as várias componentes envolvidas, associadas ao caso de uso do envio de informação entre o Provider e o Consumer:

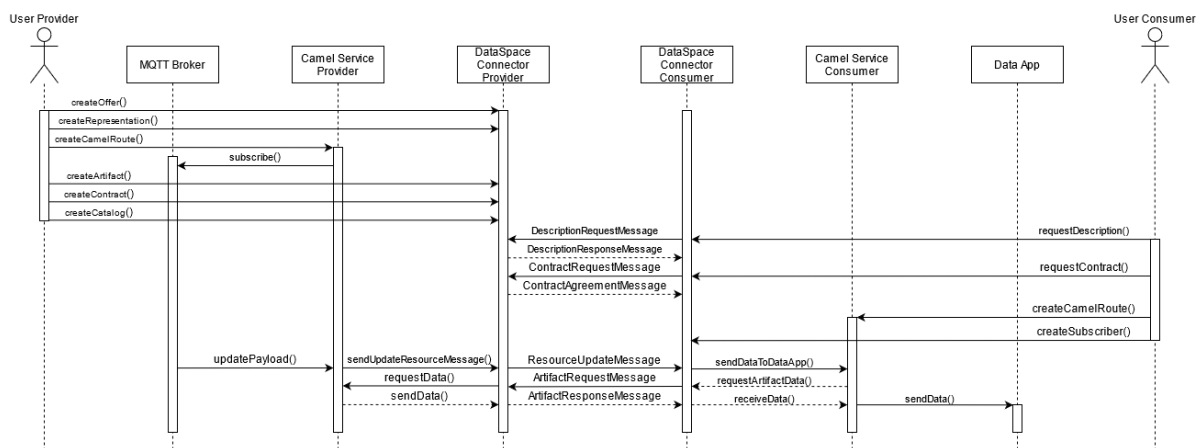


Figura 17: Diagrama de Sequência da arquitetura do protótipo.

De modo a que estas interações possam ocorrer, é necessário que se verifiquem um conjunto de pré-condições em ambas as partes (Provider e Consumer):

- **Provider:** No Provider, o utilizador (User Provider), precisa de criar um Catalog, um Resource, uma Representation e um Artifact no DataSpace Connector (Provider), para que o DataSpace Connector (Consumer) consiga descobrir e aceder aos dados. Além disso, o utilizador necessita de criar um Contract, e as respetivas rules associadas a esse Contract. Após criar o Contract e associá-lo à Resource, o utilizador, deve criar uma rota no CSP, que estará à escuta num dos tópicos do MQTT Broker. Por fim, o utilizador tem de associar o CSP ao Artifact criado, para que este consiga obter os dados armazenados no CSP.
- **Consumer:** No Consumer, o utilizador (User Consumer), necessita de realizar um conjunto de pedidos na REST API do DataSpace Connector (Consumer), que executam

as três interações apresentadas em pontos anteriores, nomeadamente, a Descrição do Provider, a Negociação de Contratos e a Partilha de Dados. Essas três interações, são utilizadas para iniciar a partilha de dados entre os dois conectores. Depois de concluídos esses pedidos, o DataSpace Connector (Consumer), sem intervenção do utilizador, cria um Resource do tipo “Requested Resource”, que contém a informação do Resource solicitado previamente, na interação da Partilha de Dados. De seguida, o utilizador deve criar uma rota no CSC, que irá fazer um conjunto de pedidos ao “Requested Resource”, para obter a última atualização dos dados e posteriormente redirecioná-los para a Data-App. Por fim, o utilizador precisa de criar uma entidade do tipo “Subscription”, que tem como objetivo associar o endpoint da rota definida no CSC ao “Requested Resource” do DataSpace Connector (Consumer).

Após cumpridas estas pré-condições, é possível iniciar a sequência de interações entre as entidades, para dar início à partilha de informação entre ambas:

1. O MQTT Broker atualiza os dados associados a um tópico;
2. A rota criada no CSP, está à escuta das mensagens provenientes do MQTT Broker e alerta o DataSpace Connector (Provider), de possíveis alterações nos dados;
3. O DataSpace Connector (Provider), ao receber o alerta, envia um conjunto de “IDS Messages”, ao DataSpace Connector (Consumer), com o objetivo de notificar as alterações que ocorreram num dos Resources.
4. O DataSpace Connector (Consumer), depois de receber uma notificação sobre a alteração nos dados de um Resource, envia um pedido para o endpoint da rota, presente no CSC.
5. Após receber esse pedido, o CSC obtém a última atualização dos dados no Resource, através do DataSpace Connector (Consumer), e redireciona-os para a DataApp.

Por fim, dá-se por terminada a sequência de ações entre as entidades, quando as DataApps recebem a última atualização dos dados provenientes do MQTT Broker, de forma automática e assíncrona, e os apresentam ao utilizador (User Consumer).

O protótipo escolhido, dispõe dos critérios definidos inicialmente e está de acordo com as restrições impostas pelo IDS. Após a escolha do protótipo, este tem de ser posto à prova perante um caso de uso real, para poder ser validado. O caso de uso utilizado para essa validação é apresentado no seguinte capítulo.

Capítulo 5

Caso de Uso

Devido à crise económico-social provocada pela pandemia e às sucessivas disrupções nas cadeias de valor em várias empresas, o interesse das empresas pelos Marketplaces voltou a despertar e com isso novas exigências e requisitos, nomeadamente no que toca a questões sobre a segurança, interoperabilidade e governação [43]. Podemos definir um Marketplace como um espaço de troca de informação entre uma ou mais empresas tipicamente do mesmo setor industrial.

Neste capítulo, propomo-nos em explorar as mais valias dos IDS no contexto da comunicação interempresarial através de Marketplaces, assentes num novo paradigma tecnológico de partilha de informação e governação em rede. O desenvolvimento de Marketplaces baseados em IDS permite abrir o leque de funcionalidades, nomeadamente:

- A possibilidade de uma integração mais ampla, ou seja, com os dispositivos/máquinas presentes no chão-de-fábrica de uma empresa, permitindo-lhes enviar informação (exemplo: estado, utilização de recursos, erros) "diretamente" para outras empresas, recorrendo plataformas IoT. Esta nova funcionalidade vem complementar os paradigmas clássicos de comunicação dos Marketplaces [43];
- Uma maior automação no fluxo de dados, isto é, permitir que exista troca de dados de forma automatizada e em tempo-real.

No contexto dos Marketplaces baseados no IDS-RAM, foi experimentada a arquitetura descrita no capítulo 4, no âmbito do projeto GREAT. A descrição do projeto e o contexto específico do caso de uso é descrito na secção seguinte.

5.1 Descrição do Problema

Atualmente a Europa representa grande parte do mercado global da produção de máquinas e equipamentos industriais [44]. Isto significa que as empresas necessitam de apostar em novas abordagens, de forma a vender e fornecer serviços pós-venda e de apoio ao cliente online, para conseguirem alcançar mercados distantes sem gastarem recursos desnecessariamente. O Market 4.0 ¹, consiste num projeto europeu onde o seu principal objetivo é desenvolver uma plataforma online europeia, para a venda de equipamento industrial e de um conjunto alargado de serviços associados, enquanto testa e valida a sua importância perante fluxos B2B.

O projeto GREAT ² conta com a participação do INESC TEC ³ e corresponde a um caso de uso do projeto Market 4.0. No projeto GREAT são apresentados três objetivos significantes:

- O **Primeiro Objetivo**, consiste no desenho e na integração com o Market 4.0 de dois modelos digitais, que representam um conjunto de máquinas físicas, utilizadas para o corte e dobra de metais. Trata-se portanto de um objetivo mais redirecionado com o conceito de "Digital Twin" [45].
- O **Segundo Objetivo**, corresponde ao desenvolvimento de uma aplicação para o Market 4.0, que permite às empresas da área metalúrgica, configurar os seus equipamentos através de um catálogo de máquinas, componentes e extensões, de acordo com os requisitos do seu produto final.
- O **Terceiro Objetivo**, consiste em obter os dados sobre os ativos industriais ⁴ das empresas na área da metalúrgica, recorrendo a serviços data-driven, e disponibilizá-los às empresas fornecedoras desses ativos, através de um serviço que complementa a aplicação desenvolvida no objetivo anterior. No âmbito deste objetivo deverão existir serviços data-driven inovadores, que se dedicam a enviar dados sobre o estado dos ativos industriais, em tempo real, com o objetivo de permitir às empresas cliente obter esses dados, para ajudar a definir planos de manutenção sem criar disrupções na cadeia de valor.

O caso de uso utilizado para a validação dos cenários propostos enquadra-se no terceiro objetivo do projeto GREAT. Cujas arquitetura genérica se apresenta na figura 18.

¹<http://market40.eu/>

²<https://www.inesctec.pt/pt/projetos/great>

³<https://www.inesctec.pt/>

⁴Os ativos industriais correspondem às máquinas/dispositivos presentes no chão de fábrica de uma empresa.

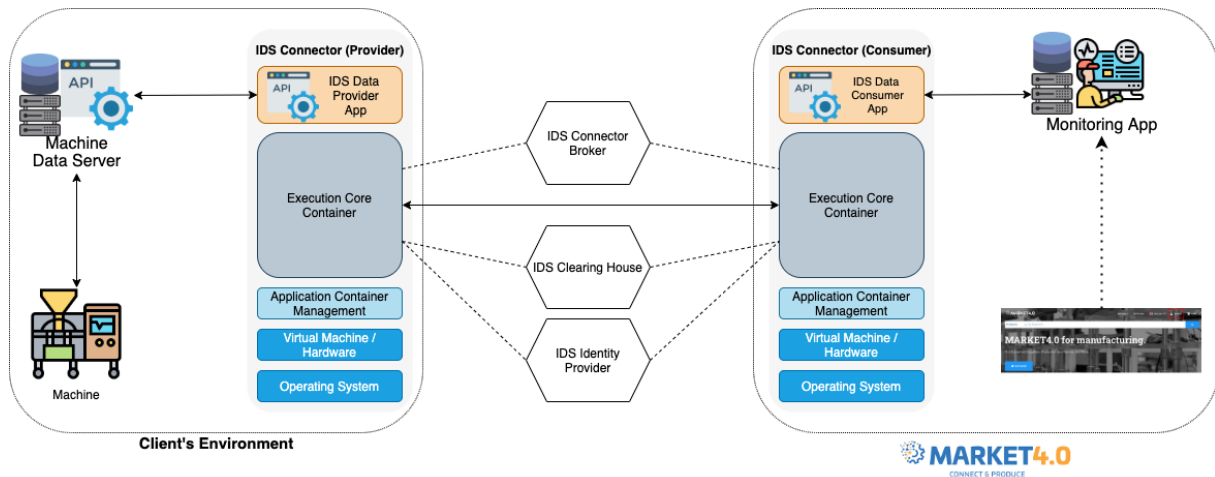


Figura 18: Arquitetura do terceiro objetivo do projeto GREAT.

A partir da figura 18, podemos observar dois ambientes, a empresa cliente e o Marketplace "Market 4.0". A empresa cliente, possui vários ativos industriais nomeadamente, as máquinas utilizadas para processar metais, que enviam constantemente os seus dados para um Servidor de Dados. O Servidor de Dados, através de um conector IDS, consegue enviar os dados recebidos para o Market 4.0. No lado do Market 4.0, existe também um conector IDS, que recebe os dados do servidor e redireciona-os para uma aplicação de monitorização. Por fim, a aplicação de monitorização, através de uma interface gráfica, permite aos utilizadores visualizarem os dados recebidos, referentes aos ativos industriais da empresa cliente.

5.2 Arquitetura da Solução

Com base na revisão de literatura e na experiência adquirida durante o desenvolvimento dos vários protótipos e cenários no âmbito dos Marketplaces, foi possível desenhar a solução, apresentada na figura 19, e desenvolver os seus componentes de modo a cumprir com as exigências impostas pelo terceiro objetivo do projeto GREAT.

Considerando que este caso de uso é a comunicação não só da fronteira da empresa A para a fronteira da empresa B, mas sim do chão de fábrica da empresa A para uma outra empresa. Existe a necessidade de haver comunicação em tempo real e da integração com diferentes sistemas. Com base nestas necessidades, verifica-se o caso de uso como adequado para experimentar a arquitetura desenhada. A arquitetura da solução, apresentada na figura 19, corresponde ao teste da arquitetura desenhada, no caso de uso. Ao aplicarmos os conceitos do IDS na solução apresentada, podemos atribuir o papel de Provider ao Client's Environment e o papel de Consumer ao Market 4.0.

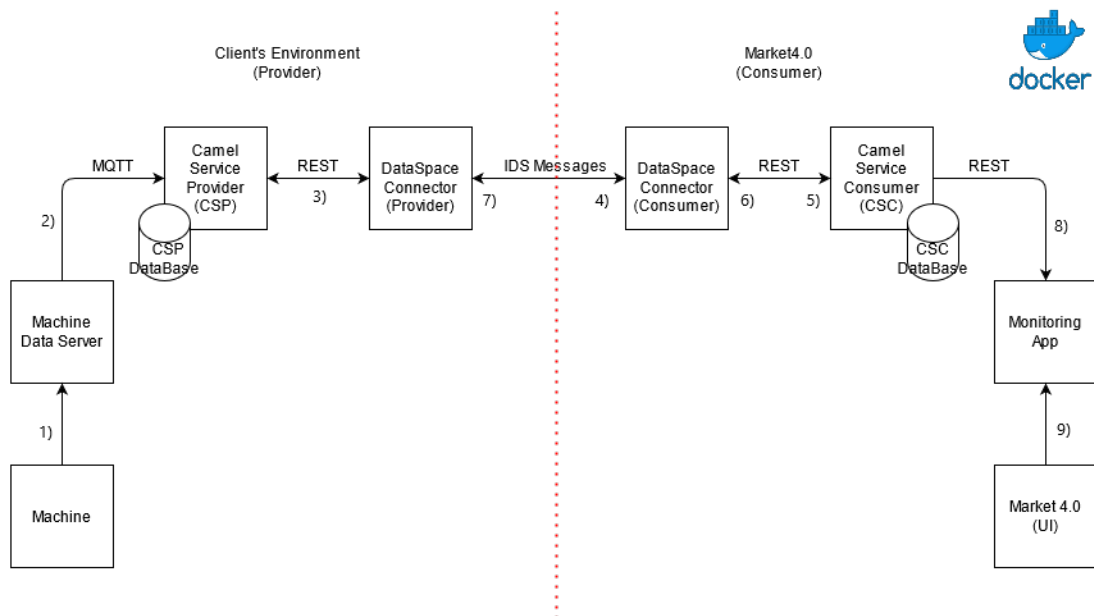


Figura 19: Arquitetura da solução apresentada para o terceiro objetivo.

De modo a facilitar a compreensão da solução, foram atribuídos números a todas as interações entre as componentes presentes na arquitetura, e cujo papel se descreve de seguida.

1. Os ativos industriais (Machine) enviam os seus dados (ex.: estado da produção, possíveis erros, utilização de recursos) para o Machine Data Server, que consiste num MQTT Broker. Para efeitos demonstrativos foi utilizado o MQTT Box para simular o envio das mensagens para o broker, como mostra a imagem 20. Concretamente a imagem representa uma mensagem "dummy" com o conteúdo "Industrial Demo Data" que será publicada no tópico "test-topic" do MQTT Broker;

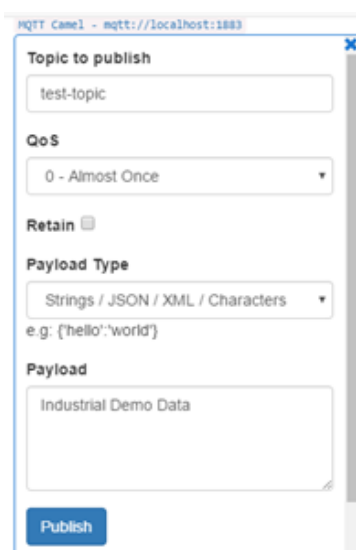


Figura 20: Exemplo de uma mensagem no MQTT Box.

2. O Camel Service Provider, está à escuta dos tópicos definidos no Machine Data Server. Deste modo, sempre que um dos tópicos é atualizado o Camel Service Provider recebe a mensagem e escreve-a na sua base de dados. A seguinte imagem da figura 21, apresenta a mensagem enviada previamente, recebida pelo Camel Service Provider.



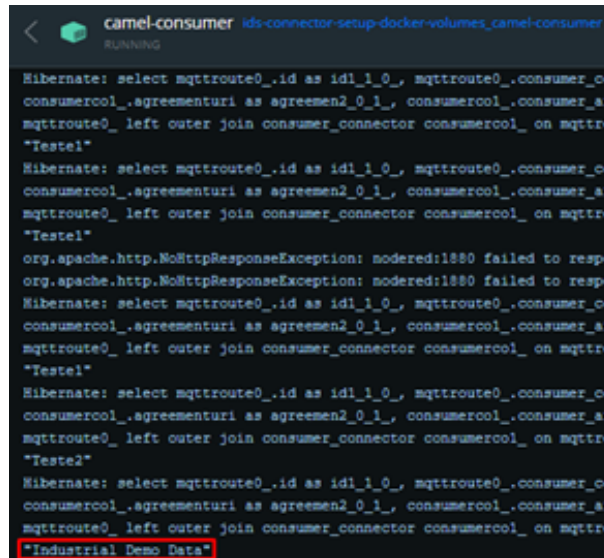
```
camel-          ids-connector-setup-docker-volumes_camel-
provider        provider
Running

providercol_.id as id1_0_0_, providercol_.password as password2
as recipien4_0_0_, providercol_.resource_id as resource5_0_0_,
join provider_connector providercol_ on mqtroute0_.provider_co
Hibernate: update routes set broker_payload=?, broker_topic=?,
Hibernate: select mqtroute0_.id as id1_1_0_, mqtroute0_.broke
mqtroute0_.brokerurl as brokerurl4_1_0_, mqtroute0_.is_active
providercol_.id as id1_0_1_, providercol_.password as password2
as recipien4_0_1_, providercol_.resource_id as resource5_0_1_,
join provider_connector providercol_ on mqtroute0_.provider_co
Industrial Demo Data
Hibernate: select mqtroute0_.id as id1_1_0_, mqtroute0_.broke
mqtroute0_.brokerurl as brokerurl4_1_0_, mqtroute0_.is_active
providercol_.id as id1_0_1_, providercol_.password as password2
as recipien4_0_1_, providercol_.resource_id as resource5_0_1_,
join provider_connector providercol_ on mqtroute0_.provider_co
Hibernate: select mqtroute0_.id as id1_1_1_, mqtroute0_.broke
mqtroute0_.brokerurl as brokerurl4_1_1_, mqtroute0_.is_active
providercol_.id as id1_0_0_, providercol_.password as password2
as recipien4_0_0_, providercol_.resource_id as resource5_0_0_,
join provider_connector providercol_ on mqtroute0_.provider_co
Hibernate: update routes set broker_payload=?, broker_topic=?,
Hibernate: select mqtroute0_.id as id1_1_0_, mqtroute0_.broke
mqtroute0_.brokerurl as brokerurl4_1_0_, mqtroute0_.is_active
providercol_.id as id1_0_1_, providercol_.password as password2
as recipien4_0_1_, providercol_.resource_id as resource5_0_1_,
join provider_connector providercol_ on mqtroute0_.provider_co
```

Figura 21: Exemplo de uma mensagem recebida no Camel Service Provider.

3. O Camel Service Provider, depois de armazenar a ultima atualização dos dados na sua base de dados, envia um pedido para o DataSpace Connector (Provider).
4. O DataSpace Connector (Provider), após receber o pedido do Camel Service Provider, cria e envia uma notificação sobre as possíveis atualizações nos dados recebidos, ao Data Space Connector (Consumer).
5. O DataSpace Connector (Consumer), depois de receber a notificação do DataSpace Connector (Provider), envia um pedido para a rota no Camel Service Consumer, previamente definida.

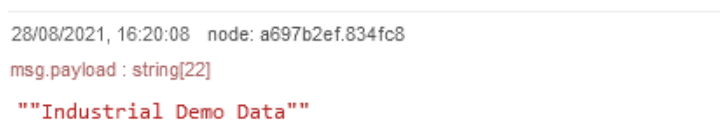
6. O Camel Service Consumer, ao receber o pedido do conector, faz uma solicitação da última atualização dos dados ao Provider através do DataSpace Connector (Consumer). A seguinte imagem da figura 22 apresenta os dados recebidos pelo Camel Service Consumer.



```
camel-consumer | ids-connector-setup-docker-volumes_camel-consumer
RUNNING
Hibernate: select mqtroute0_.id as id1_1_0_, mqtroute0_.consumer_co
consumerool_.agreementuri as agreemen2_0_1_, consumerool_.consumer_art
mqtroute0_ left outer join consumer_connector consumerool_ on mqtrou
"Teste1"
Hibernate: select mqtroute0_.id as id1_1_0_, mqtroute0_.consumer_co
consumerool_.agreementuri as agreemen2_0_1_, consumerool_.consumer_art
mqtroute0_ left outer join consumer_connector consumerool_ on mqtrou
"Teste1"
org.apache.http.NoHttpResponseException: modered:1880 failed to respon
org.apache.http.NoHttpResponseException: modered:1880 failed to respon
Hibernate: select mqtroute0_.id as id1_1_0_, mqtroute0_.consumer_co
consumerool_.agreementuri as agreemen2_0_1_, consumerool_.consumer_art
mqtroute0_ left outer join consumer_connector consumerool_ on mqtrou
"Teste1"
Hibernate: select mqtroute0_.id as id1_1_0_, mqtroute0_.consumer_co
consumerool_.agreementuri as agreemen2_0_1_, consumerool_.consumer_art
mqtroute0_ left outer join consumer_connector consumerool_ on mqtrou
"Teste2"
Hibernate: select mqtroute0_.id as id1_1_0_, mqtroute0_.consumer_co
consumerool_.agreementuri as agreemen2_0_1_, consumerool_.consumer_art
mqtroute0_ left outer join consumer_connector consumerool_ on mqtrou
"Industrial Demo Data"
```

Figura 22: Exemplo de uma mensagem recebida no Camel Service Consumer.

7. O DataSpace Connector (Provider), recebe a solicitação dos dados e comunica com o Camel Service Provider, de modo a obter os dados que estão armazenados na sua base de dados, para depois os enviar à entidade que os solicitou.
8. A Monitoring App, possui um endpoint ao qual o Camel Service Consumer consegue aceder e enviar os dados recebidos. Para efeitos demonstrativos foi utilizado o NodeRed para simular a Monitoring App, como mostra a seguinte imagem da figura 23.



```
28/08/2021, 16:20:08 node: a697b2ef.834fc8
msg.payload : string[22]
"Industrial Demo Data"
```

Figura 23: Exemplo de uma mensagem recebida no NodeRed.

9. Por fim, os dados recebidos na Monitoring App, podem ser acedidos e visualizados a partir da aplicação Market 4.0.

A arquitetura da solução apresentada, cumpre com todos os requisitos impostos pelo terceiro objetivo do projeto GREAT, observando-se o seguinte:

- A presença de comunicação entre o ponto 1 e o ponto 9 da figura 19;
- O paradigma "Publish and Subscribe" foi implementado com o auxílio do Serviço Camel;
- A possibilidade de integrar aplicações de terceiros (ex.: MQTT Broker, NodeRed), através do Serviço Camel;
- A comunicação em "tempo real".

Por fim, depois de terminada a experiência e verificada a comunicação entre os dois conectores IDS, utilizando o serviço Camel conforme desenhado na arquitetura. A solução, foi validada por elementos do projeto GREAT e containerizada, para poder ser utilizada no piloto. Depois da containerização da solução desenvolvida, cada componente possui o seu próprio container, onde se encontram as dependências e configurações necessárias para a sua execução. A solução desenvolvida possui duas versões, uma versão para efeitos demonstrativos e uma versão para ser aplicada no projeto GREAT. As configurações necessárias para iniciar ambas as versões, estão presentes no apêndice A deste relatório.

Capítulo 6

Conclusões e Trabalho Futuro

Após o estudo de diversas abordagens, utilizadas num contexto de partilha de informação interorganizacional, explorou-se os International Data Spaces como abordagem emergente para troca de dados entre empresas, visto que oferecem uma rede flexível, descentralizada e onde a segurança, governação dos dados e a confiança entre os participantes são requisitos obrigatórios. Assim sendo, a IDSA, apresentou um modelo de arquitetura de referência, com o propósito de facilitar a sua implementação e responder a estes requisitos.

A abordagem baseada em IDS irá contribuir para uma maior democratização tecnológica no contexto da partilha de informação interorganizacional, pelo que a sua exploração no âmbito dos Marketplaces é pertinente e promissora. Face aos objetivos propostos pelo projeto GREAT, foi projetada uma solução onde foi utilizado o International Data Space, de modo a possibilitar a partilha de dados em tempo-real e sem comprometer com a segurança e o seu conteúdo. A arquitetura cumpriu com os objetivos do piloto e foi validada pelos elementos do projeto. Considerando a abordagem qualitativa, o trabalho discutido neste documento cumpriu o seu propósito.

6.1 Reflexão Crítica

Apesar dos IDSs serem a opção mais adequada para a partilha de informação interorganizacional no contexto da Indústria 4.0, a sua adoção por pequenas e médias empresas conta com diversas limitações, devido à sua complexidade, e como não compensa os investimentos necessários continuam a não ser uma opção viável para este tipo de empresas.

Este trabalho contribui com mais um caso para aumentar a base de conhecimento no domínio da partilha de informação 4.0. Parece-nos um instrumento para transferência de conhecimento da academia para a indústria.

É ainda importante referir, que o estado de maturidade atual desta tecnologia ainda não está no ponto ideal para a adoção em massa pelas empresas. No entanto, conclui-se ser uma tecnologia promissora face aos desafios atuais no domínio da troca de dados uniformizada e segura entre organizações.

6.2 Trabalho Futuro

Num futuro próximo existe espaço para aperfeiçoar o IDS Connector, as suas componentes e em geral o International Data Space, de modo a eliminar as atuais limitações com o objetivo de aumentar a sua compatibilidade com pequenas e médias empresas. Além disso, investir na implementação de outras componentes do IDS-RAM, uma vez que este trabalho foca essencialmente nos conectores IDS, e na experimentação da arquitetura em outros contextos e casos de uso. A longo prazo e com o surgimento da Indústria 5.0, que complementa a atual Indústria 4.0 ao focar-se na sustentabilidade e ao criar uma indústria Human-Centric, é possível considerar os IDSs como uma abordagem a seguir para este novo paradigma de partilha de informação interorganizacional.

Bibliografia

- [1] A. Rojko, "Industry 4.0 Concept: Background and Overview," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 11, p. 77, July 2017.
- [2] M. M. Pereira, "Vertical and horizontal integration in the media sector and EU competition law " The ICT and Media Sectors within the EU Policy Framework "," *Access*, p. 12, 2003.
- [3] D. Horváth and R. Z. Szabó, "Driving forces and barriers of Industry 4.0: Do multinational and small and medium-sized companies have equal opportunities?," *Technological Forecasting and Social Change*, vol. 146, pp. 119–132, sep 2019.
- [4] R. S. Kumar and S. Pugazhendhi, "Information Sharing in Supply Chains: An Overview," *Procedia Engineering*, vol. 38, pp. 2147–2154, 2012.
- [5] S. Bhartiya and D. Mehrotra, "Exploring Interoperability Approaches and Challenges in Healthcare Data Exchange," in *Smart Health* (D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, D. Zeng, C. C. Yang, V. S. Tseng, C. Xing, H. Chen, F.-Y. Wang, and X. Zheng, eds.), vol. 8040, pp. 52–65, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [6] Becca McCarthy, "EDI History," 2013.
- [7] M. Kantor and A. Prabhakar, "FIPS 161-2 - (EDI), Electronic Data Interchange," 1996.
- [8] K. V. G, "BUSINESS TO BUSINESS (B2B) AND BUSINESS TO CONSUMER (B2C) MANAGEMENT," *International Journal of Computer & Technology*, 2012.
- [9] OpenText, "Understanding the benefits of EDI (Electronic Data Interchange)," 2017.
- [10] "EDI Web | EDI Solutions | Electronic Data Interchange | Solutions | Edicom."
- [11] J.-M. Nurmilaakso, "EDI, XML and e-business frameworks: A survey," *Computers in Industry*, vol. 59, pp. 370–379, apr 2008.

- [12] S. SME, “Vendor Managed Inventory (VMI): Three Steps in Making It Work | SCM | Supply Chain Resource Cooperative (SCRC) | North Carolina State University,” Mar. 2003.
- [13] M. J. Franklin, “Dataspaces: Progress and Prospects,” in *Dataspace: The Final Frontier* (A. P. Sexton, ed.), Lecture Notes in Computer Science, (Berlin, Heidelberg), pp. 1–3, Springer, 2009.
- [14] H. Banaee, E. Schaffernicht, and A. Loutfi, “Data-driven Conceptual Spaces: Creating Semantic Representations For Linguistic Descriptions Of Numerical Data,” *Journal of Artificial Intelligence Research*, vol. 63, pp. 691–742, Nov. 2018.
- [15] Carmine F. Greco, “Shared Data Spaces,” 1997.
- [16] M. Murth and E. Kühn, “Knowledge-based interaction patterns for semantic spaces,” in *2010 International Conference on Complex, Intelligent and Software Intensive Systems*, (Los Alamitos, CA, USA), pp. 1036–1043, IEEE Computer Society, feb 2010.
- [17] M. L. Zeng, “Knowledge Organization Systems (KOS),” *KNOWLEDGE ORGANIZATION*, vol. 35, no. 2-3, pp. 160–182, 2008.
- [18] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau, “Extensible Markup Language (XML) 1.0 (Fifth Edition),” Nov. 2008.
- [19] Dan Brickley, R.V. Guha, and Brian McBride, “RDF Schema 1.1,” Feb. 2014.
- [20] Deborah L. McGuinness and Frank van Harmelen, “OWL Web Ontology Language Overview,” Feb. 2004.
- [21] P. Eckert, “Communities of practice,” *Encyclopedia of language and linguistics*, vol. 2, no. 2006, pp. 683–685, 2006.
- [22] C. Bizer, T. Heath, and T. Berners-Lee, “Linked Data - The Story So Far:,” *International Journal on Semantic Web and Information Systems*, vol. 5, pp. 1–22, July 2009.
- [23] J. Umbrich, M. Karnstedt, J. X. Parreira, A. Polleres, and M. Hauswirth, “Linked data and live querying for enabling support platforms for web dataspace,” in *2012 IEEE 28th International Conference on Data Engineering Workshops*, pp. 23–28, 2012.
- [24] Open Data Institute, “Open data means business: UK innovation across sectors and regions,” 2015.
- [25] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, “DBpedia: A Nucleus for a Web of Open Data,” pp. 722–735, 2007.

- [26] A. Rojko, "Industry 4.0 concept: Background and overview," *International Journal of Interactive Mobile Technologies*, vol. 11, no. 5, pp. 77–90, 2017.
- [27] M. Paiva, A. Vasconcelos, and B. Fragoso, "Using enterprise architecture to model a reference architecture for industry 4.0," *ICEIS 2020 - Proceedings of the 22nd International Conference on Enterprise Information Systems*, vol. 2, pp. 709–716, 2020.
- [28] Bill Lydon, "RAMI 4.0 Reference Architectural Model for Industrie 4.0," 2019.
- [29] Colin Koh, "Home > industries 4.0 > Articles > The Industry 4.0 Framework: Idea, Tool, & Guide (Part 2) RAMI 4.0 Model Industry 4.0 The Industry 4.0 Model Framework: Idea, Tool, & Guide (Part 2)," 2020.
- [30] S. R. Bader, M. Maleshkova, and S. Lohmann, "Structuring reference architectures for the industrial Internet of Things," *Future Internet*, vol. 11, no. 7, pp. 1–23, 2019.
- [31] Otto, B, Steinbuß, S, Teuscher, A, and Lohmann, S, "IDS Reference Architecture Model 3.0," *International Data Spaces Association*, 2019.
- [32] Basbayandur, Onur, Böhmer, Martin, Dr.-Ing. Cirullies, Jan, Dabrowski, Agatha, Hilpert, Thomas, Jakob, Sabine, Klukas, Achim, Kraft, Volker, Leveling, Jens, Noll, Frederik, Dr. Sprenger, Philipp, Wang, Gong, Weißenberg, Norbert, Zajac, Markus, Zrenner, Johannes, and Florian Bächle, "Challenges and Potentials of a Logistic Data Space," 2019.
- [33] S. Bader, J. Pullmann, C. Mader, S. Tramp, C. Quix, A. W. Müller, H. Akyürek, M. Böckmann, B. T. Imbusch, J. Lipp, S. Geisler, and C. Lange, "The International Data Spaces Information Model – An Ontology for Sovereign Exchange of Digital Content," pp. 176–192, 2020.
- [34] E. Andreas, J. Christian, K. Christian, B. Fabian, B. Gerd, B. Pascal, N. Ralf, and B. Sebastian, "Usage Control in International Data Spaces," *International Data Spaces Association Anna-Louisa-Karsch-Str. 2 10178 Berlin Germany*, 2019.
- [35] Fraunhofer, "IDS Trusted Connector," 2020.
- [36] Fraunhofer, "Dataspace Connector: Manual and Documentation," 2021.
- [37] FIWARE Foundation, "FIWARE," 2012.
- [38] Peter Salhofer, "A Case-Study on Implementing Smart Application with FIWARE," *Evaluating the FIWARE Platform*, p. 9, 2018.

- [39] FIWARE Foundation, "FIWARE MARKETPLACE."
- [40] DE-CIX Management GmbH, G. Eggers, B. Fondermann, G. G. GmbH, B. Maier, K. Ottradovetz, J. Pfrommer, R. Reinhardt, H. Rollin, A. Schmieg, S. Steinbuß, P. Trinius, A. Weiss, C. Weiss, and S. Wilfling, "GAIA-X: Technical Architecture," p. 52, 2020.
- [41] B. Otto, A. Rubina, A. Eitel, A. Teuscher, A. M. Schleimer, C. Lange, D. Stingl, E. Loukipoudis, G. Brost, G. Böge, H. Pettenpohl, J. Langkau, J. Gelhaar, K. Mitani, M. Hupperz, M. Huber, N. Jahnke, R. Brandstädter, S. Wessel, and S. Bader, "Gaia-x and ids," 01 2021.
- [42] H. Gregor and Bobby Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Professional; 1st edition (October 10, 2003), 2003.
- [43] S. Aguiar, R. Pinto, J. Reis, and G. M. Gonçalves, "On the development of a cyber-physical industrial marketplace," p. 13, 2018.
- [44] INESC TEC, "First European digital marketplace for the metal and plastics industry resorts to INESC TEC technology," 2021.
- [45] E. Negria, L. Fumagallia, and M. Macchi, "A Review of the Roles of Digital Twin in CPS-based Production Systems," *27th International Conference on Flexible Automation and Intelligent Manufacturing*, p. 10, 2017.

Apêndice A

Guia de Configurações

O guia de configurações, contém uma descrição passo-a-passo das configurações necessárias para dar início à troca de mensagens entre o Provider e o Consumer, presentes na arquitetura da solução apresentada neste trabalho.

A.1 Pré-requisitos

Antes de dar início a qualquer configuração, é necessário verificar se os seguintes pré-requisitos estão presentes no sistema.

1. Docker Desktop (<https://www.docker.com/products/docker-desktop>);
2. Postman (<https://www.postman.com/>);
3. MQTT Box (<https://chrome.google.com/webstore/detail/mqttbox/kaajoficamnjjhkeomgfljpicifbkaf>).

Nota: Para iniciar os containers e criar os respetivos volumes é necessário executar o script `create-docker-volumes.sh` e fazer `docker compose` do ficheiro `docker-compose.yml`. Ambos os ficheiros encontram-se na pasta `ids-connector-setup-docker-volumes`.

A.2 Configurações do Provider

1. Entrar no Swagger UI do DataSpace Connector Provider (<http://localhost:8081/api/docs>) e colocar as seguintes credenciais:

- **Username:** admin
- **Password:** password

2. Criar uma nova Offer:

- **URL:** POST <http://localhost:8081/api/offers>

- **Body:**

```
{
  "title": "Sample Resource",
  "description": "This is an example resource containing
    industrial data.",
  "keywords": [
    "machine",
    "data",
    "sample"
  ],
  "publisher": "https://example.org/",
  "language": "EN",
  "licence": "http://example.org/licenses/1.0/",
  "sovereign": "https://example.org/",
  "endpointDocumentation": "https://example.org/docs"
}
```

- **Response:**

```
...
"self": {
  "href": http://localhost:8081/api/offers/${OFFERID}
}
...
```

Nota: O "OFFERID" representa o Id da Offer criada.

3. Criar uma nova Route no Camel Service Provider utilizando o Postman:

- **URL:** POST <http://localhost:8090/MQTTRoutes/addRoute>

- **Body:**

```
{
  "brokerURL": "tcp://mosquitto:1883",
  "brokerTopic": "test-topic",
  "brokerPayload": "",
  "providerConnector": {
    "providerUrl": "http://provider:8081",
    "recipientUrl": "http://consumer:8080/api/ids/data",
    "resourceId": "http://provider:8081/api/offers/${
      OFFERID}",
    "username": "admin",
    "password": "password"
  }
}
```

- **Response:**

```
{
  "headers": {},
  "body": {
    "id": ${CSPROUTEID},
    "brokerURL": "tcp://mosquitto:1883",
    "brokerTopic": "test-topic",
    "brokerPayload": "",
    "providerConnector": {
      "id": ${CSPROUTEID},
      "providerUrl": "http://provider:8081",
      "recipientUrl": "http://consumer:8080/api/ids/
        data",
      "resourceId": "http://provider:8081/api/offers/${
        OFFERID} ",
      "username": "admin",
      "password": "password"
    }
  }
}
```

```

    },
    "isActive": false
  },
  "statusCodeValue": 200,
  "statusCode": "OK"
}

```

Nota: O "CSPROUTEID" representa o Id da Route criada no Camel Service Provider.

4. Ativar a Route criada através do Postman:

- **URL:** POST <http://localhost:8090/MQTTRoutes/activateRoute>
- **Body:** CSPROUTEID
- **Response:**

```

{
  "headers": {},
  "body": "Route ${CSPROUTEID} Activated!",
  "statusCodeValue": 200,
  "statusCode": "OK"
}

```

5. Criar um Artifact no conector e definir o accessUrl com o endpoint da Route criada previamente:

- **URL:** POST <http://localhost:8081/api/artifacts>
- **Body:**

```

{
  "title": "Industrial Sample",
  "accessUrl": "http://camel-provider:8090/MQTTRoutes/
    getPayload/${CSPROUTEID}/%2F**"
}

```

- **Response:**

```

...
"self": {

```

```
    "href": "http://localhost:8081/api/artifacts/${ARTIFACTID}
      }"
}
...

```

6. Criar uma Representation:

- **URL:** POST <http://localhost:8081/api/representations>

- **Body:**

```
{
  "title": "Industrial Representation",
  "mediaType": "TEXT",
  "language": "EN"
}
```

- **Response:**

```
...
"self": {
  "href": "http://localhost:8081/api/representations/${
    REPRESENTATIONID}"
},
...

```

Nota: O "REPRESENTATIONID" representa o Id da Representation criada no conector.

7. Associar o Artifact, criado no ponto 5, à Representation:

- **URL:** POST [http://localhost:8081/api/representations/\\${REPRESENTATIONID}/artifacts](http://localhost:8081/api/representations/${REPRESENTATIONID}/artifacts)

- **Body:**

```
[
  "https://localhost:8081/api/artifacts/${ARTIFACTID}"
]
```

- **Response:** Lista dos Artifacts adicionados à Representation.

8. Adicionar a Representation, criada no ponto 6, à Offer:

- **URL:** POST `http://localhost:8081/api/offers/${OFFERID}/representations`

- **Body:**

```
[  
  "http://localhost:8081/api/representations/${  
    REPRESENTATIONID}"  
]
```

- **Response:** Lista das Representations adicionadas à Offer.

9. Criar um Contract:

- **URL:** POST `http://localhost:8081/api/contracts`

- **Body:**

```
{  
  "title": "Industrial Contract",  
  "start": "2021-06-16T18:36:19.386Z",  
  "end": "2022-06-16T18:36:19.386Z"  
}
```

- **Response:**

```
...  
"self": {  
  "href": "http://localhost:8081/api/contracts/${CONTRACTID}"  
},  
...
```

Nota: O "CONTRACTID" representa o Id do Contract criado.

10. Criar uma Rule para aceder aos dados:

- **URL:** POST <http://localhost:8081/api/rules>

- **Body:**

```
{
  "title": "Provide Access",
  "value": "{\ "@type\":"ids:Permission\","@id\":"https://w3id.org/idsa/autogen/permission/fb0f770a-fa1a-4147-a346-90825343f182\","ids:target\":null,"ids:description\":[{\ "@value\":"provide-access\","@type\":"http://www.w3.org/2001/XMLSchema#string"}],\ "ids:action\":[{\ "properties\":null,\ "@id\":"idsc:USE"}],\ "ids:preDuty\":null,\ "ids:assetRefinement\":null,\ "ids:postDuty\":null,\ "ids:constraint\":null,\ "ids:title\":[{\ "@value\":"Allow Data Usage\","@type\":"http://www.w3.org/2001/XMLSchema#string"}],\ "ids:assignee\":null,\ "ids:assigner\":null}"
}
```

- **Response:**

```
...
"self": {
  "href": "http://localhost:8081/api/rules/${ACCESSRULEID}"
},
...
```

Nota: O "ACCESSRULEID" representa o Id da regra de acesso criada.

11. Criar uma Rule para o envio de notificações:

- **URL:** POST <http://localhost:8081/api/rules>
- **Body:**

```
{
  "title": "Notification",
  "value": "{\ "@type\":"ids:Permission", "@id\":"https://w3id.org/idsa/autogen/permission/d6c1b372-b705-4d08-b0fe-1d474306ca6e", "ids:target":null, "ids:description\":[{\ "@value\":"usage-notification", "@type\":"http://www.w3.org/2001/XMLSchema#string"}], "ids:action\":[{\ "properties\":null, "@id\":"idsc:USE"}], "ids:preDuty":null, "ids:assetRefinement":null, "ids:postDuty\":[{\ "@type\":"ids:Duty", "@id\":"https://w3id.org/idsa/autogen/duty/17473450-6a78-4f55-9b57-c253cc91a74f", "ids:target":null, "ids:description\":null, "ids:action\":[{\ "properties\":null, "@id\":"idsc:NOTIFY"}], "ids:assetRefinement\":null, "ids:constraint\":[{\ "@type\":"ids:Constraint", "@id\":"https://w3id.org/idsa/autogen/constraint/9da1690e-d786-473b-8871-97fc9995f877", "ids:rightOperandReference\":null, "ids:leftOperand\":[{\ "properties\":null, "@id\":"idsc:ENDPOINT"}], "ids:rightOperand\":[{\ "@value\":"https://localhost:8080/api/ids/data", "@type\":"xsd:anyURI"}], "ids:unit\":null, "ids:pipEndpoint\":null, "ids:operator\":[{\ "properties\":null, "@id\":"idsc:DEFINES_AS"}]}, "ids:title\":null, "ids:assignee\":null, "ids:assigner\":null}], "ids:constraint\":null, "ids:title\":[{\ "@value\":"Example Usage Policy", "@type\":"http://www.w3.org/2001/XMLSchema#string"}], "ids:assignee\":null, "ids:assigner\":null}"
}
```

- **Response:**

```
...  
"self": {  
  "href": "http://localhost:8081/api/rules/${  
    NOTIFICATIONRULEID}"  
},  
...
```

Nota: O "NOTIFICATIONRULEID" representa o Id da regra criada para o envio de notificações.

12. Associar as Rules ao Contract criado no ponto 9:

- **URL:** POST `http://localhost:8081/api/contracts/${CONTRACTID}/rules`

- **Body:**

```
[  
  "http://localhost:8081/api/rules/${ACCESSRULEID}",  
  "http://localhost:8081/api/rules/${NOTIFICATIONRULEID}"  
]
```

- **Response:** Lista das Rules adicionadas ao Contract.

13. Associar o Contract à Offer:

- **URL:** POST `http://localhost:8081/api/offers/${OFFERID}/contracts`

- **Body:**

```
[  
  "http://localhost:8081/api/contracts/${CONTRACTID}"  
]
```

- **Response:** Lista dos Contracts adicionados à Offer.

14. Criar um Catalog:

- **URL:** POST <http://localhost:8081/api/catalogs>

- **Body:**

```
{  
  "title": "Sample Catalog",  
  "description": "Catalog Description"  
}
```

- **Response:**

```
...  
"self": {  
  "href": "http://localhost:8081/api/catalogs/${CATALOGID}"  
},  
...
```

Nota: O "CATALOGID" representa o Id do Catalog criado.

15. Associar a Offer ao Catalog criado:

- **URL:** POST [http://localhost:8081/api/catalogs/\\${CATALOGID}/offers](http://localhost:8081/api/catalogs/${CATALOGID}/offers)

- **Body:**

```
[  
  "http://localhost:8081/api/offers/${OFFERID}"  
]
```

- **Response:** Lista de Offers adicionadas ao Catalog.


```

    {
      "@type": "ids:Permission",
      "@id": "http://provider:8081/api/rules/${
        NOTIFICATIONRULEID}"
    }
  ],
  ...

```

Nota: A descrição retorna um atributo "ids:contractOffer" que contém um array de Permissions, que têm de ser enviados no ponto seguinte. No final de cada Permission é necessário adicionar o atributo "ids:target": "https://localhost:8080/api/artifacts/\${ARTIFACTID}".

4. Fazer um pedido ao Provider para negociar um Contract:

- **URL:** POST `http://localhost:8080/api/ids/contract?recipient=http%3A%2F%2Fprovider%3A8081%2Fapi%2Fids%2Fdata&resourceIds=http%3A%2F%2Fprovider%3A8081%2Fapi%2Foffers%2F${OFFERID}&artifactIds=http%3A%2F%2Fprovider%3A8081%2Fapi%2Fartifacts%2F${ARTIFACTID}&download=false`
- **Body:**

```

[
  {
    "@type": "ids:Permission",
    "@id": "http://provider:8081/api/rules/${
      ACCESSRULEID}"
    "ids:target": "http://provider:8081/api/
      artifacts/${ARTIFACTID}"
  },
  {
    "@type": "ids:Permission",
    "@id": "http://provider:8081/api/rules/${
      NOTIFICATIONRULEID}"
    "ids:target": "http://provider:8081/api/
      artifacts/${ARTIFACTID}"
  }
]

```

- **Response:**

```
...
"remoteId": http://provider:8081/api/agreements/${
  AGREEMENTID}
...
```

Nota: O "AGREEMENTID" corresponde ao Id do acordo aceite por ambos os conectores.

5. (Opcional) Fazer um primeiro pedido de dados ao Artifact, criado automaticamente no lado do Consumer, para testar a conexão:

- **URL:** GET `http://consumer:8080/api/artifacts/${CONSUMERARTIFACTID}/data/**?download=true&agreementUri=http%3A%2F%2Fprovider%3A8081%2Fapi%2Fagreements%2F${AGREEMENTID}`

Nota: O "CONSUMERARTIFACTID" representa o Id do Artifact criado automaticamente no conector Consumer.

6. Criar uma nova Route no Camel Service Consumer através do Postman:

- **URL:** POST `http://localhost:8091/MQTTRoutes/addRoute`

- **Body:**

```
{
  "dataAppURL": "http://nodered:1880/endpoint",
  "consumerConnector": {
    "consumerArtifactURI": "http://consumer:8080/api/artifacts/${CONSUMERARTIFACTID}",
    "agreementURI": "http://provider:8081/api/agreements/${AGREEMENTID}",
    "params": {},
    "username": "admin",
    "password": "password"
  }
}
```

- **Response:**

```
{
  "headers": {},
  "body": {
    "id": "${CSCROUTEID}",
    "dataAppURL": "http://nodered:1880/endpoint",
    "consumerConnector": {
      "id": "${CSCROUTEID}",
      "consumerArtifactURI": "http://consumer:8080/api/artifacts/${CONSUMERARTIFACTID}",
      "agreementURI": "http://provider:8081/api/agreements/${AGREEMENTID}",
      "params": {},
      "username": "admin",
      "password": "password"
    }
  },
  "statusCodeValue": 200,
  "statusCode": "OK"
}
```

Nota: O "CSCROUTEID" representa o Id da Route criada no Camel Service Consumer.

7. Criar um Subscriber:

- **URL:** POST `http://localhost:8080/api/subscriptions`

- **Body:**

```
{
  "title": "Industrial Subscriber",
  "description": " Notify on Update",
  "target": "http://consumer:8080/api/artifacts/${
    CONSUMERARTIFACTID}",
  "location": "http://camel-consumer:8091/MQTTRoutes/
    sendDataToDataApp/${CSCROUTEID}",
  "subscriber": "http://camel-consumer:8091",
  "pushData": false
}
```

- **Response:**

```
...
"self": {
  "href": "http://localhost:8080/api/subscriptions/${
    SUBSCRIBERID} "
}
...
```

Nota: O "SUBSCRIBERID" corresponde ao Id do Subscriber criado no conector.

8. Associar o Subscriber criado ao respetivo Consumer Artifact:

- **URL:** POST `http://localhost:8080/api/artifacts/${CONSUMERARTIFACTID}/subscriptions`

- **Body:**

```
[
  "http://localhost:8080/api/subscriptions/${SUBSCRIBERID} "
]
```

- **Response:** Lista de Subscribers associados ao Artifact.