TÉCNICO LISBOA

Enterprise Integration

Kick off & *Introduction*

Prof. Sérgio Guerreiro
Sergio.guerreiro@tecnico.ulisboa.pt

Department of Computer Science and Engineering
Instituto Superior Técnico / Universidade de Lisboa
INESC-ID
URL: http://www.inesc-id.pt
Rua Alves Redol, 9
1000-029 Lisboa
Portugal

Faculty

Faculty

Sérgio Guerreiro Senior Lecturer
sergio.guerreiro@tecnico.ulisboa.pt

Flávio Martins
flavio.f.martins@tecnico.ulisboa.pt

Major IE 2025 Goals

- To learn what enterprise integration is
- To understand that integration and cloud are fundamental parts of the digital economy
- To have a practical experience with the “cloud” integration technologies

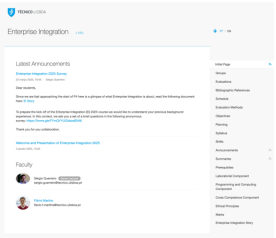
In the end, to understand the information systems of digital (big) companies

Detailed Course Program

The course aims at providing a coherent structure of integration topics that can be found in different parts of the ACM/AIS IS 2010 curriculum, such as “Enterprise Systems” and “Application Development”

- Evolution of information systems. Functions. Evolution. Service.
- Introduction to integration platforms. Messages. Transformations. Schemas. Ports. Adapter. Orchestration. Business rule.
- Messaging systems. Transactions. Correlations. Platforms.
- Message brokers. Publish-subscribe. Streaming. Asynchronous.
- Adapters. 3-tier. Methods. Database. Files. User interface. Dynamic invocations. Web services.
- Micro-services, services and SOA. Applications. Composition. Catalog. Business processes. Service design. Human workflows.
- Service orchestrations. Flow. Decisions. Loops. Events. Exceptions. Transactions.
- Inter-organizational integration. Electronic data exchange. Supply chain management. E-commerce. Negotiation. Internet of things.

Hyperautomation is a transversal concern in all those aspects. How to do it faster and better?



TÉCNICO LISBOA

Study artifacts

- **Academic page for course @ Fénix:** https://fenix.tecnico.ulisboa.pt/disciplinas/SEI36/2024-2025/2-semestre
- **Q&A @ Moodle:** https://moodle.dei.tecnico.ulisboa.pt/course/view.php?id=6273
- **AWS Academy classroom for each sprint -> invitation sent for sprint 1 (using your fenix registered email)**
  - **SPRINT 1** https://awsacademy.instructure.com/courses/117013
  - **SPRINT 2** https://awsacademy.instructure.com/courses/117014
- **Slides + Project + Scenarios + tutorials are already available @ Fénix**

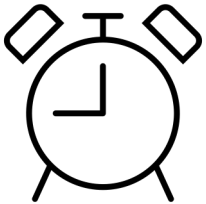
Class **dynamics**

This is a **hands-on** course using **flipped learning**:

1. Understand the IE concepts **at concept classes**
2. Execute the tutorial, **at home**
3. Watch the movie tutorial, if needed, **at home**
4. Deploy the scenarios and adapt them, **at home**
5. Bring questions/problems to apply the tutorial to the **LAB** and **project** classes

Weekly schedule

	Seg 5/5	Ter 5/6	Qua 5/7	Qui 5/8	Sex 5/9
07:00					
08:00					
		08:30 - 10:00 L F8	08:30 - 10:30 T EA3	08:30 - 10:00 L V1.25	
09:00					
	09:30 - 11:30 T EA3				
10:00		10:00 - 11:30 L F8		10:00 - 11:30 L V1.12	
11:00					



The slots are organized as **concept classes** and **lab classes** and **project classes**

				Content	Tutorial Guidelines	Tutorial movies	Project Deliverables
28/04/2025	Week 1	EA3	Concepts	00 - Intro / 00 - Enterprise Integration			
29/04/2025		F8 + F8	LAB	Kafka in the cloud	P1-Kafka-in-AWSAcademy.pdf	EC2-VM_Creation_and_Accessing Kafka_Installation_EC2 Using_Kafka_EC2	
30/04/2025		EA3	Concepts	01 - MoM - Kafka - Streams			
01/05/2025		V1.25 + V1.12	*Holiday: Labor Day*				
05/05/2025	Week 2	EA3	LAB	Kafka in the cloud (cont.) Terraform	P1-Kafka-in-AWSAcademy.pdf P2-HyperAutomation-TERRAFORM.pdf	EC2-VM_Creation_and_Accessing Kafka_Installation_EC2 Using_Kafka_EC2	
06/05/2025		F8 + F8	LAB	Kafka clustering Kafka Streams	P4-Kafka-Streams-AWS.pdf P3-Kafka-Distributed.pdf	KafkaCluster-Configuration	
07/05/2025		EA3	Concepts	02 - SOA & Microservices & Reactive & Quarkus			
08/05/2025		V1.25 + V1.25	LAB	Quarkus	P6-Quarkus.pdf		
12/05/2025	Week 3	EA3	LAB	Quarkus with Docker (introductory)	P6-Quarkus.pdf		
13/05/2025		F8 + F8	LAB	Quarkus: REST & DB & Swagger	P6-Quarkus.pdf P5-RDS-database.pdf		
14/05/2025		EA3	Evaluation	Revision + MAP45			
15/05/2025		V1.25 + V1.12	Project	Sprint 1 support			Saturday 17/5/2025 12h Sprint 1 (15%)
19/05/2025	Week 4	EA3	Concepts	03 - Containers & Cloud			
20/05/2025		F8 + F8	LAB	AWS LAMBDA	P8-Lambda-AWS.pdf	Develop Deploy and Invoke a Lambda microservice	
21/05/2025		EA3	Concepts	04 - Processes			
22/05/2025		V1.25 + V1.12	LAB	Sprint 1 Patterns feedback			
26/05/2025	Week 5	EA3	LAB	Camunda platform	P7-BPMN-CAMUNDA.pdf Scenario 1	Camunda-Overview A full example of a executable BPMN collaboration in Camunda engine	
27/05/2025		F8 + F8	LAB	Business process design & implementation Business process Integration & DMN	P7-BPMN-CAMUNDA.pdf Scenario 1	Camunda-Overview A full example of a executable BPMN collaboration in Camunda engine	
28/05/2025		EA3	Concepts	05 - API management and Gateway systems			
29/05/2025		V1.25 + V1.12	LAB	Authentication & credentials APIs Deployment & management	P9-KONG.pdf Scenario2	Kong APIGateway Configuration and Invocation	
02/06/2025	Week 6	EA3	Concepts	05 - Identity management & B2B & Other aspects			
03/06/2025		F8 + F8	LAB	Authentication & credentials / APIs Deployment & management (cont.) Ollama deployment in EC2	P9-KONG.pdf Scenario2 P2-HyperAutomation-TERRAFORM.pdf	Kong APIGateway Configuration and Invocation	
04/06/2025		EA3	Evaluation	Revision + MAP45			
05/06/2025		V1.25 + V1.25	Project	Sprint 2 Support	Scenario 3		
09/06/2025	Week 7	EA3	Project	Sprint 2 Support			
10/06/2025		F8 + F8	*Holiday: Day of Portugal, Camões and Portuguese Communities*				
11/06/2025		EA3	Project	Sprint 2 support			
12/06/2025		V1.25 + V1.25	Project	Sprint 2 Support			Saturday 14/6/2025 12h Sprint 2 (25%)
	Week 8		project discussion				
			Normal Season Exam - 26/6/2025 10:30				
			Second Season Exam - 10/7/2025 15:30				
			Special Season Exam - 23/7/2025 18:00				

TÉCNICO LISBOA

Course Planning

	amount of classes
LAB	12
Concepts	7
project	5
evaluation	2

Links for tutorial movies

TÉCNICO LISBOA

(use IST ID Gsuite only for accessing)

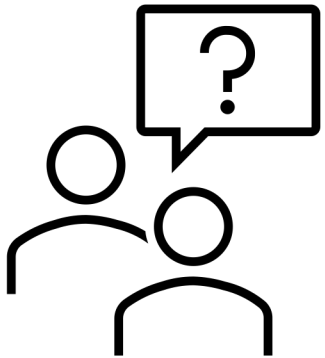
How to provision an EC2 VM?	https://drive.google.com/file/d/1GicH9SufhMnl2oCSRGjYJteNRxPFKtH-/view?usp=sharing
How to install Kafka in an EC2 VM?	https://drive.google.com/file/d/1zp_tEYVgq1o9psL7y064qMDWNoQmKuqd/view?usp=sharing
How to use Kafka in an EC2 VM?	https://drive.google.com/file/d/1h5qW2X9AM7aEVvzuMv7FhfTcXgyC7gCU/view?usp=sharing
How to enforce a Kafka Cluster?	https://drive.google.com/file/d/1da2aEJ4HywPNgCQCH4tztLRRR9wqK2t3/view?usp=sharing
How to develop, deploy, and execute an AWS Lambda service?	https://drive.google.com/file/d/1rAr27ZAZRdlw9sVCmThUNa_6zietEV5F/view?usp=sharing
What is Camunda?	https://drive.google.com/file/d/1b-j7-OaS1gWM1mJaQGbjt6vWU9L2MYme/view?usp=sharing
How to model, develop, deploy, and execute a business process using Camunda engine?	https://drive.google.com/file/d/1Yxqch1nGm7hOD3Cpz1eiiQeuzmK5Tt2h/view?usp=sharing
How to install and operate Kong in an EC2 VM?	https://drive.google.com/file/d/1vryBWG4BqgLt4cL1uUXwXC_f3Emma-vS/view?usp=sharing

Questions classes

By Zoom: https://videoconf-colibri.zoom.us/j/2785489605?pwd=SXFmNzdMZXA2TTliZ09QSFZPUXJhUT09

Send an email requesting it beforehand.

Tuesday and Thursday at 16:30



Evaluation

The final grade is given by:

NotaFinal época Normal = round(MAX(0,2 * NExameNormal + 0,40 * NMAP45 ; 0,60 * ExameNormal) + 0,40 * NProjects)

OR

NotaFinal época Recurso = round(0,60* NExameRecurso + 0,40*NProjects)

OR

NotaFinal época Especial = round(0,60* NExameEspecial + 0,40*NProjects)

Where:

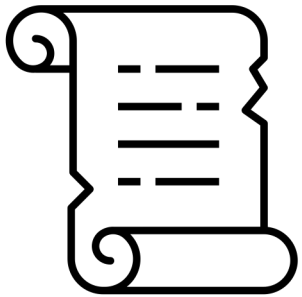
2 * NMAP45 are done during classes

NExameNormal, NMAP45, NExameRecurso and NExameEspecial requires >= 8,0 values

NProjects consists of 2 sprints (15% - 25% for each sprint)

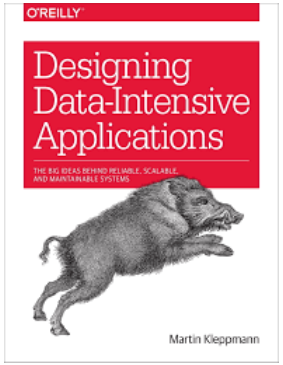
NProjects >= 8.0 values -> Project is done by groups of 2 students

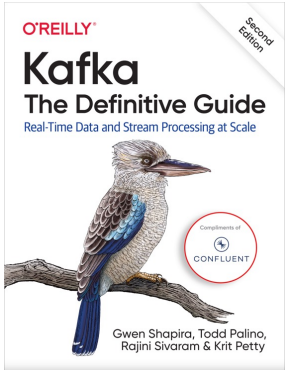
NotaFinal época Normal, NotaFinal época Recurso, NotaFinal época Especial >= 9,5 values

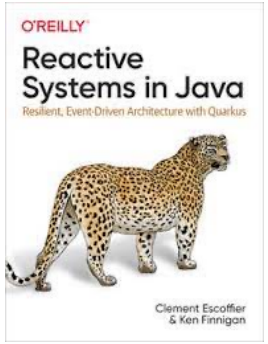


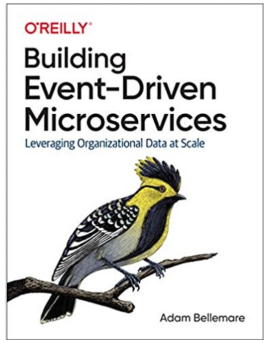
TÉCNICO LISBOA

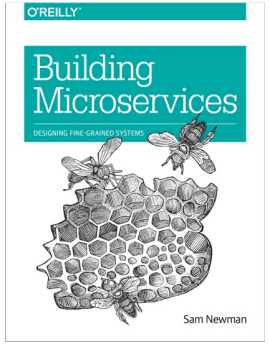
Books

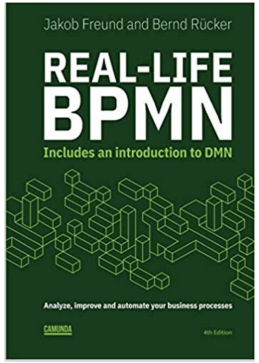


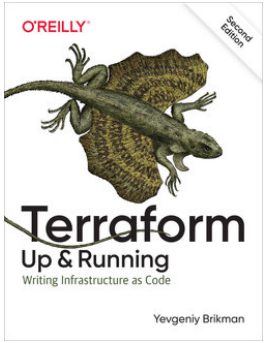


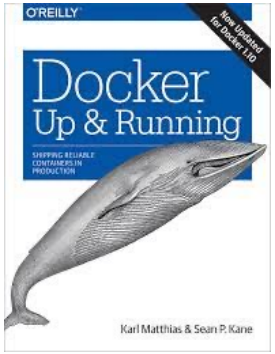


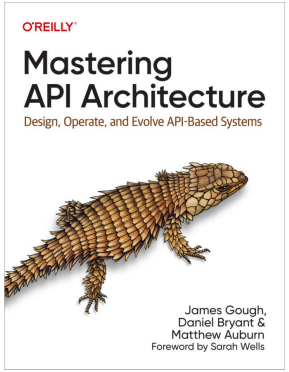












Books

Gwen Shapira, Todd Palino, Rajini Sivaram & Krit Petty (2021) Kafka: The Definitive Guide, O'Reilly Media, Inc., ISBN: 978-1-491-99065-0

Escoffier, C., Finnigan, K. (2021). Reactive Systems in Java, O'Reilly Media, Inc., ISBN: 9781492091721

Freund J., Rücker, B. (2019). Real-Life BPMN (4th edition): Includes an introduction to DMN, Camunda, ISBN: 978-1086302097

Kane, S., Matthias, K. (2018). Docker: Up & Running, 2nd Edition, O'Reilly Media, Inc., ISBN: 9781492036739

Kleppmann, M. (2017). Designing Data-Intensive Applications, O'Reilly Media, Inc., ISBN: 9781449373320

Bellemare, A. (2020). Building Event-Driven Microservices: Leveraging Organizational Data at Scale, O'Reilly Media, ISBN: 978-1492057895

Brikman, Y. (2019), Terraform: Up & Running, 2nd Edition, O'Reilly Media, Inc., ISBN: 9781492046905

Newman, S. (2021). Building Microservices, 2nd Edition. O'Reilly Media, Inc., ISBN: 9781492034025

Gough, J., Bryant, D., Auburn, M. (2023). Mastering API Architecture: Design, Operate, and Evolve API-Based Systems, O'Reilly Media, Inc., ISBN: 9781492090632

The technology that you'll be using

TÉCNICO LISBOA

Business Processes execution

API Management

MicroServices

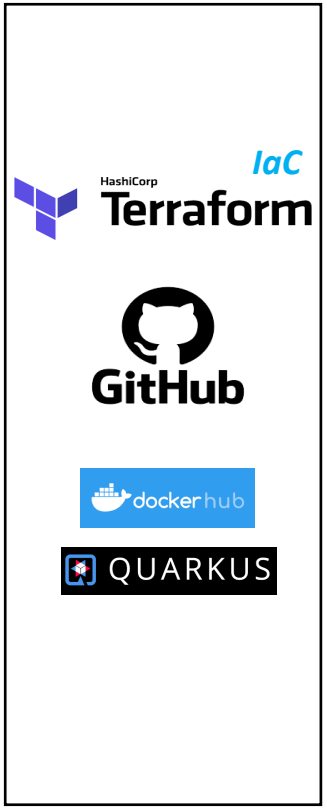
Messaging

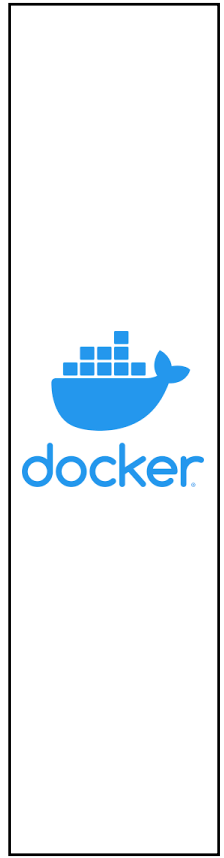
PaaS IaaS

Containerization

Hyperautomation

IaC





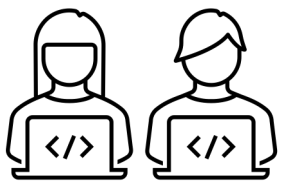
TÉCNICO LISBOA

- Windows: putty + fileZilla
- Linux or macOS: check if `ssh` and `scp` are working

Minimum tools to install on your environment

see tutorials for further details

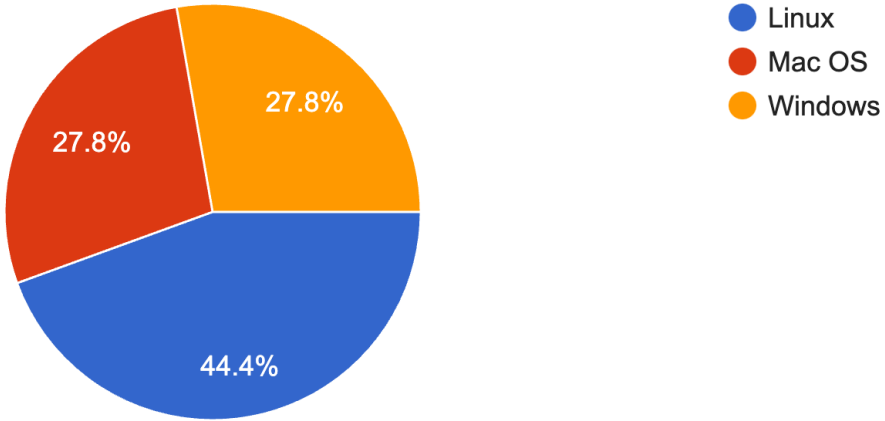
- All Operating systems:
  - VS Code https://code.visualstudio.com/
  - Eclipse (*optionally to VS Code*) https://www.eclipse.org/downloads/
  - JDK 17 https://adoptium.net/
  - Maven 3.9.9 https://maven.apache.org/download.cgi
  - Terraform https://developer.hashicorp.com/terraform/downloads
  - Docker desktop https://www.docker.com/get-started
  - Camunda-modeler (v7) https://camunda.com/download/modeler/
  - MYQSL Workbench https://dev.mysql.com/downloads/workbench/
  - Postman https://www.postman.com/downloads/



TÉCNICO LISBOA

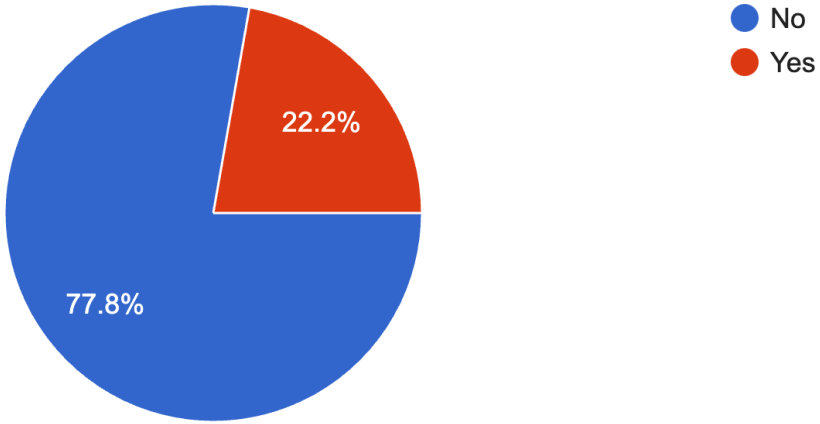
What is the Operating System that you are planning to use during EI course?

18 responses

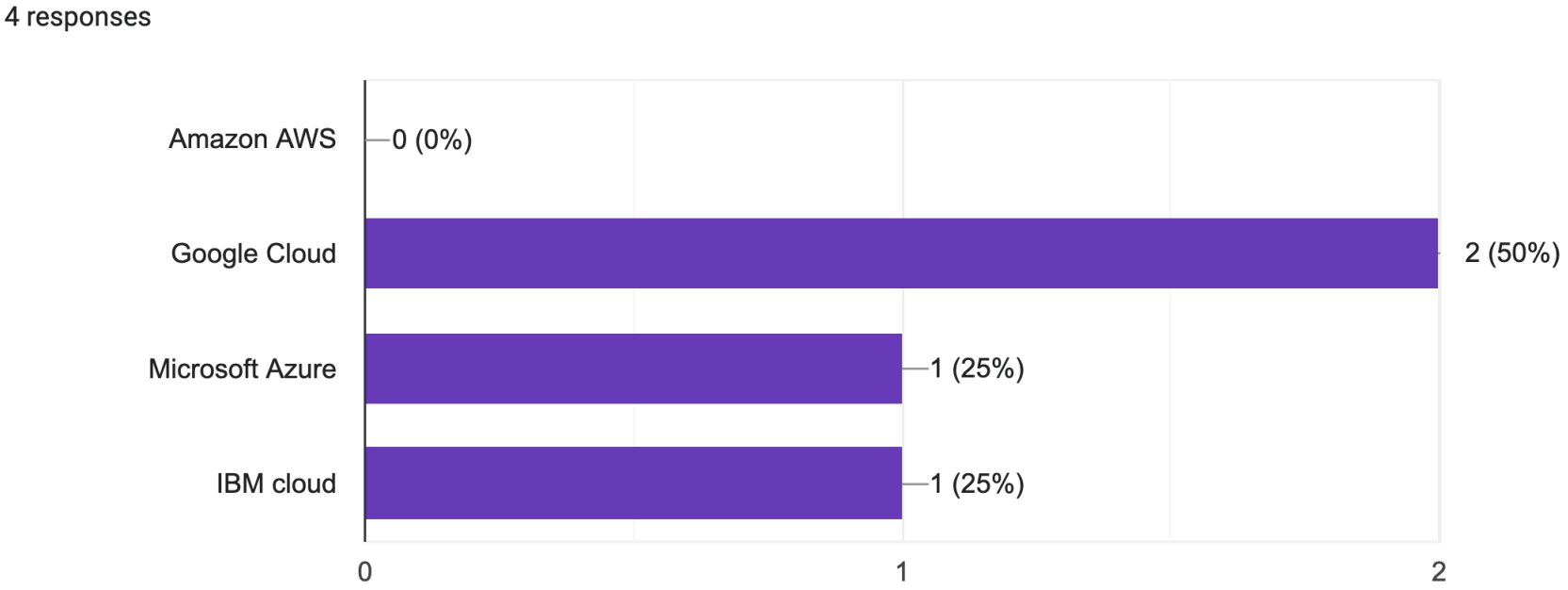


Have you ever developed and/or configured using cloud environments (e.g.: AWS, Microsoft Azure, Google cloud, other....)?

18 responses



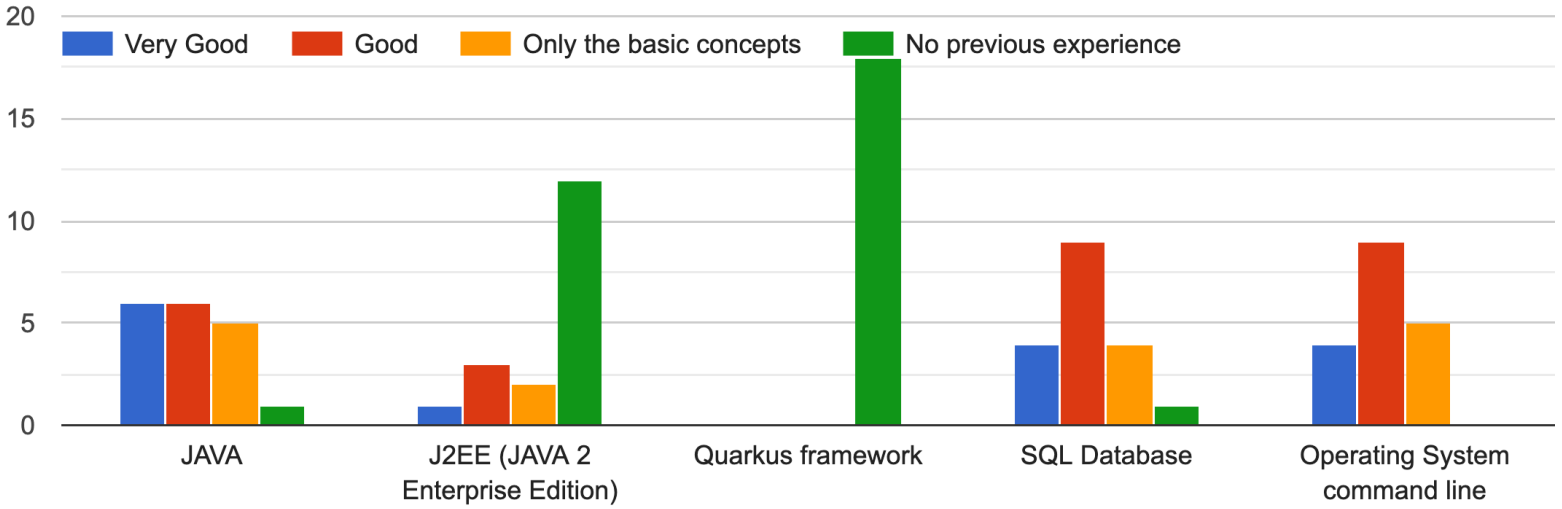
Which cloud environment are you experienced with?



TÉCNICO LISBOA

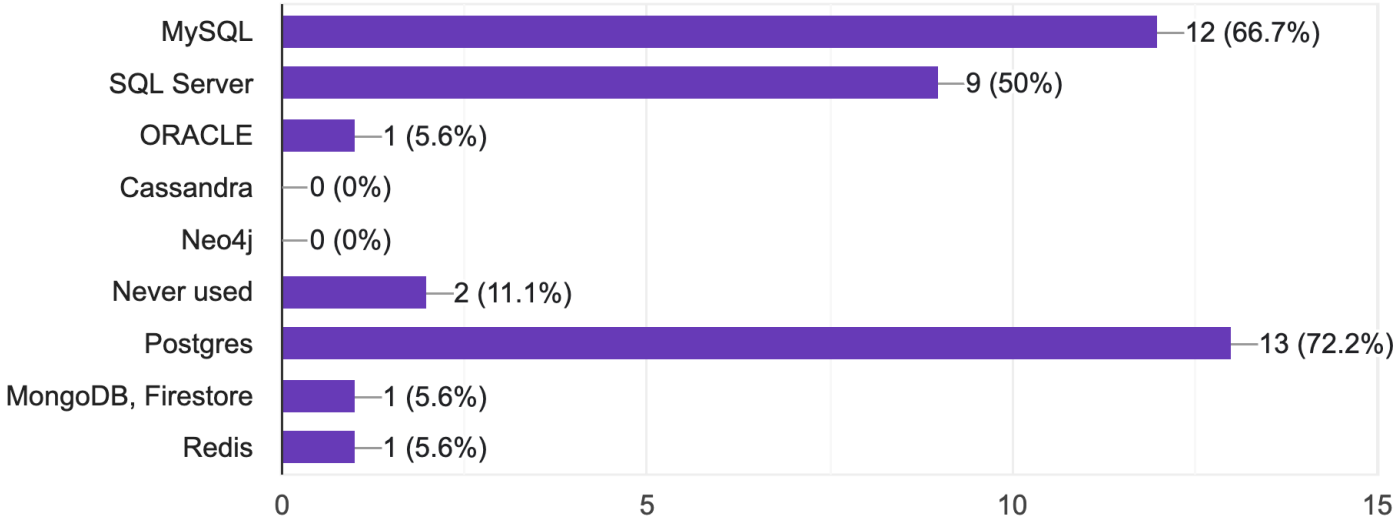
Programming background - What is your experience in the following technologies?

Choose one option per row.



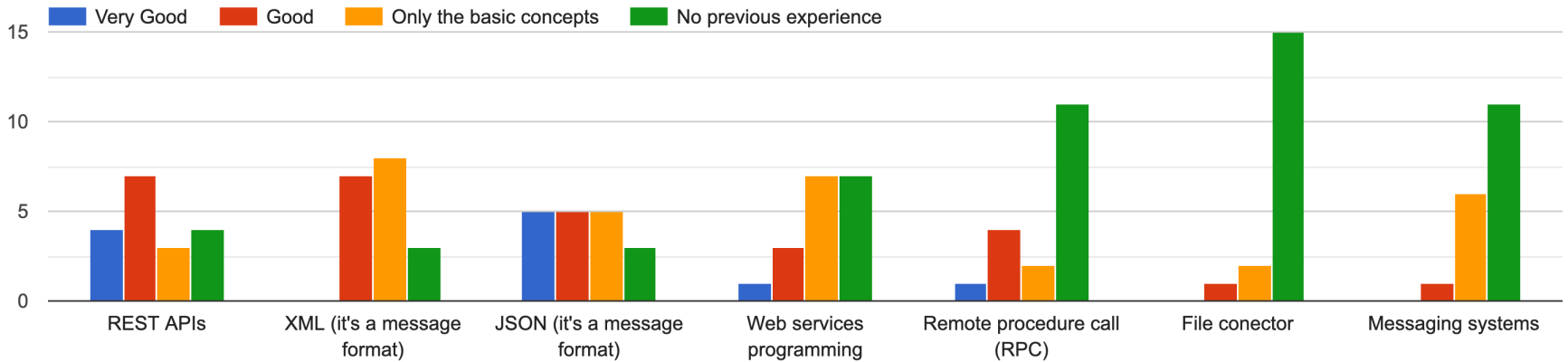
Regarding Database knowledge or experience, which technologies did you use? Choose one or more options.

18 responses



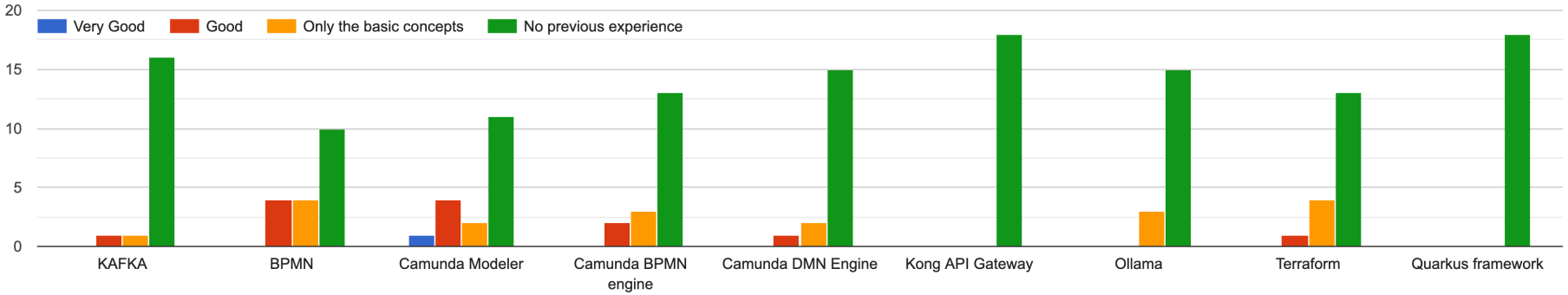
Integration using API - What is your experience in the following technologies?

Choose one option per row.



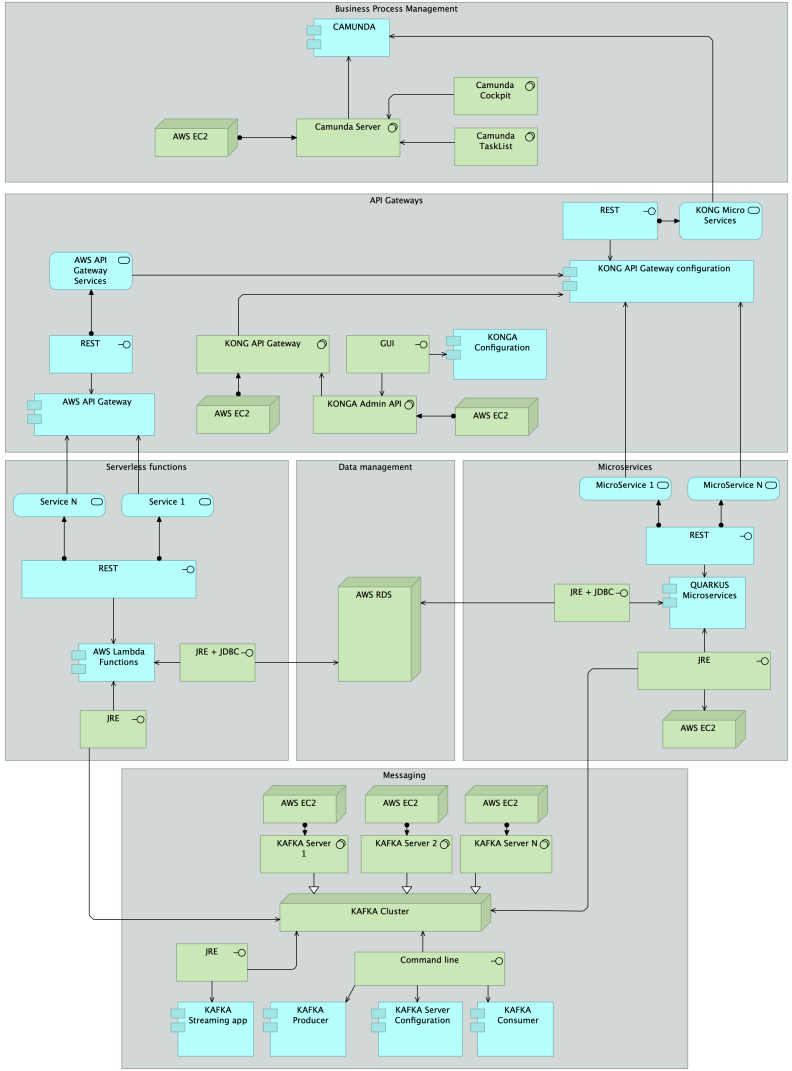
Knowledge related with other integration systems - What is your experience in the following technologies?

Choose one option per row.



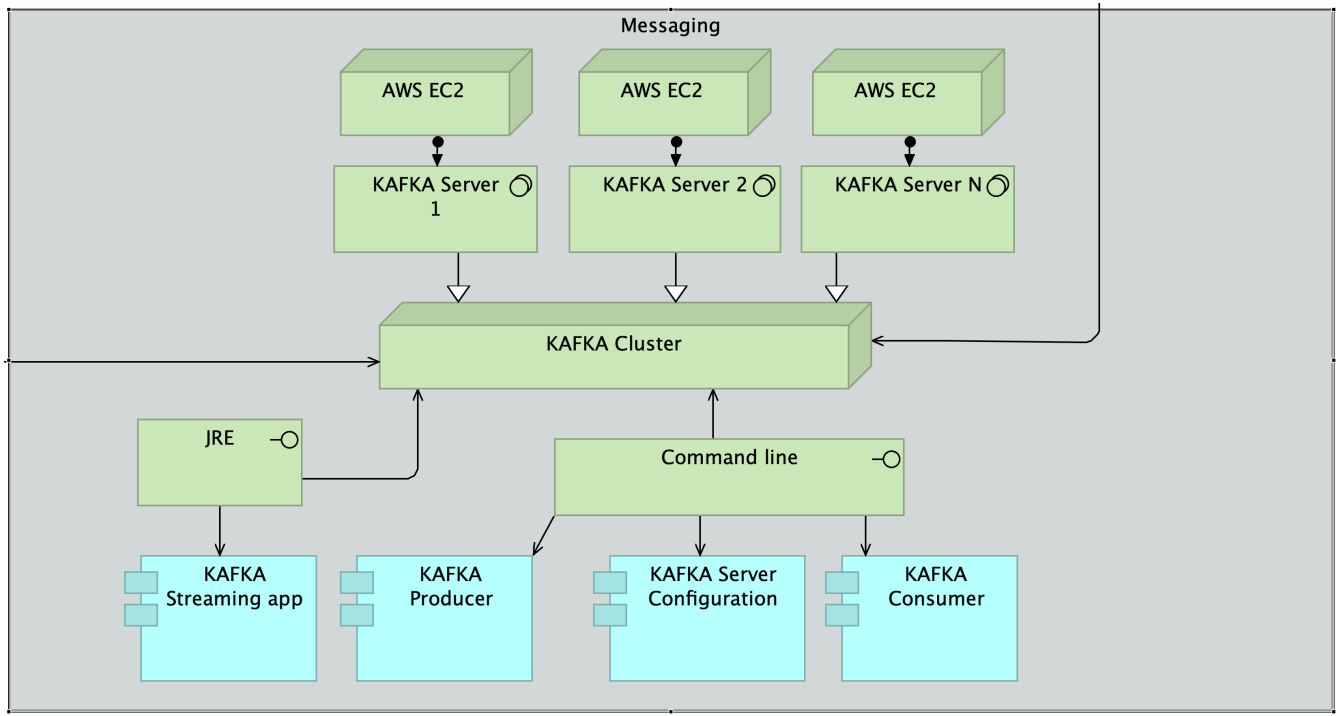
IE Technology Stack

Overall view



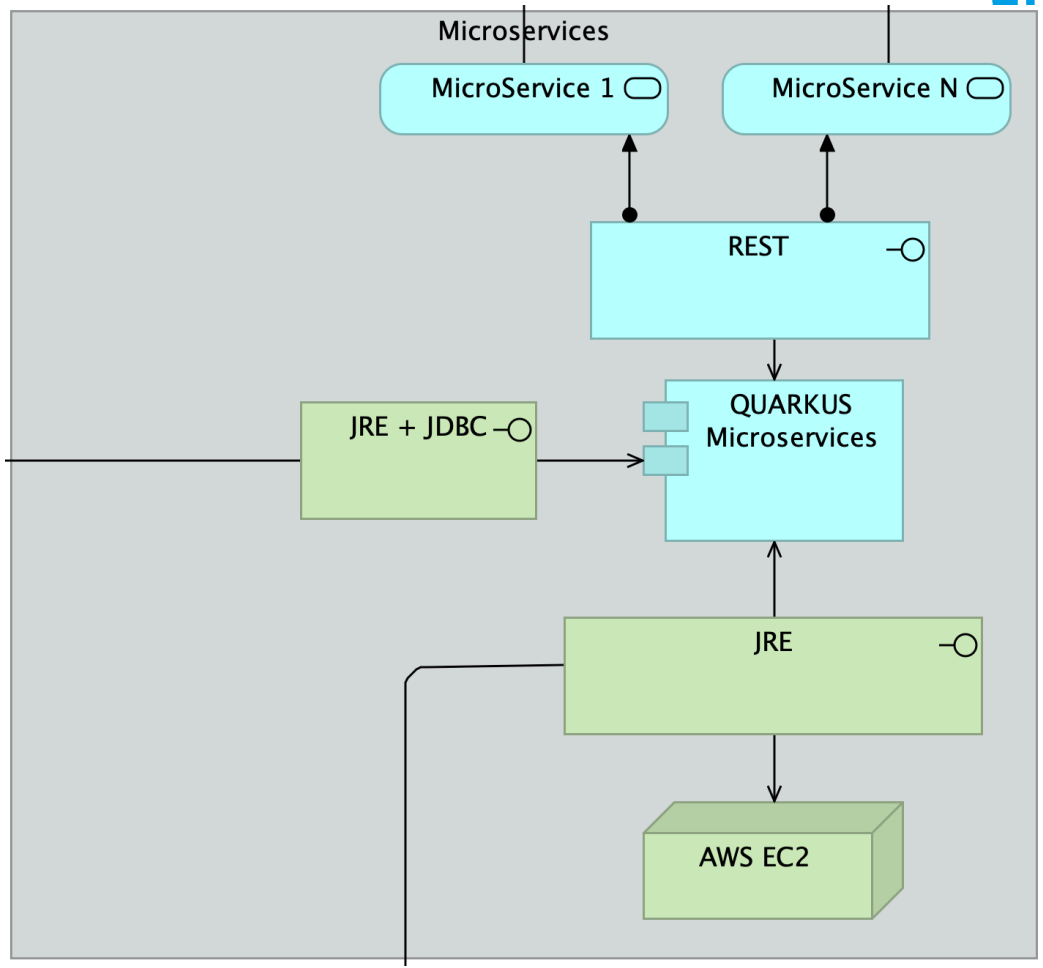
IE Technology Stack

Messaging layer



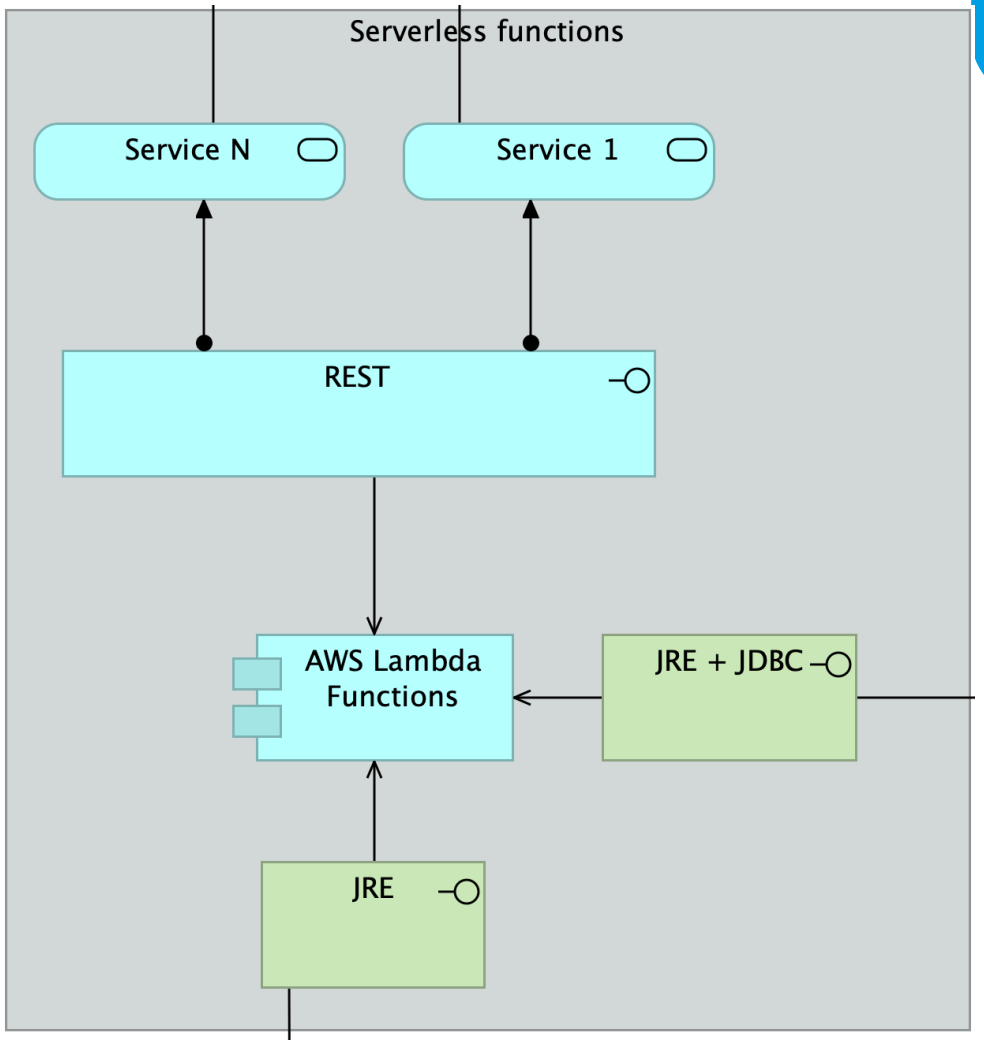
IE Technology Stack

MicroService layer



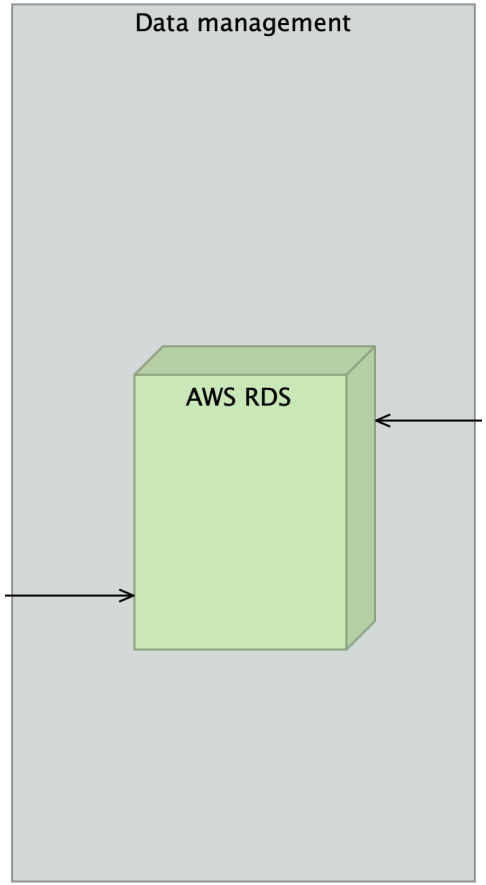
IE Technology Stack

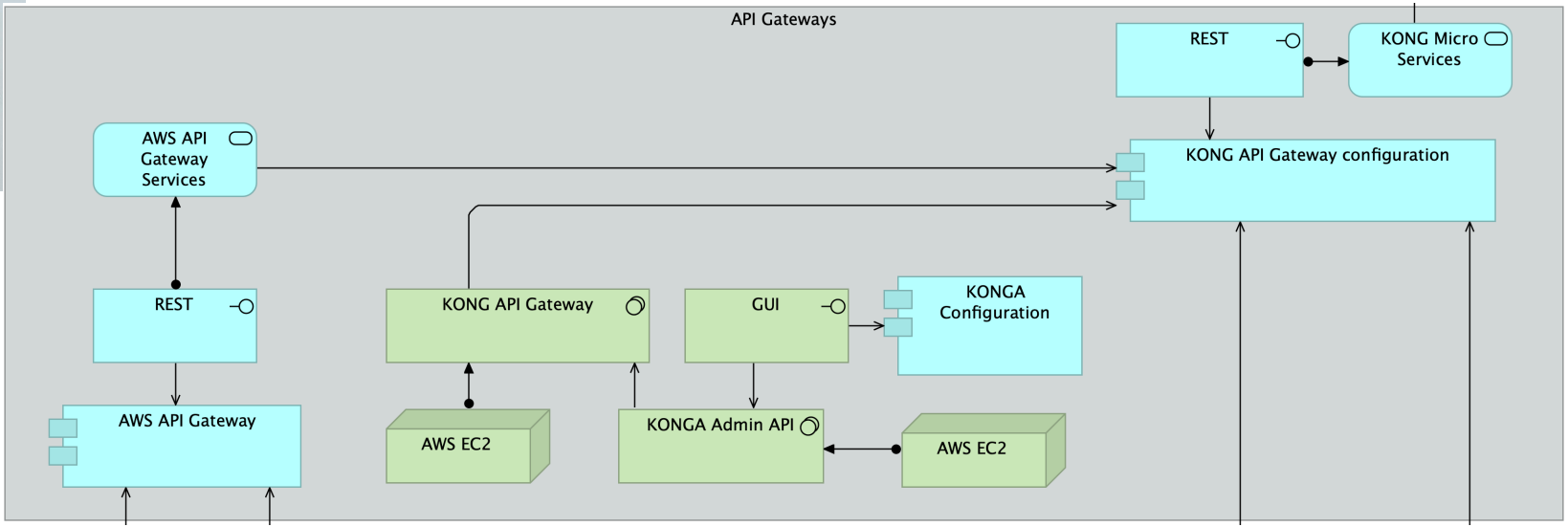
Serverless layer



IE Technology Stack

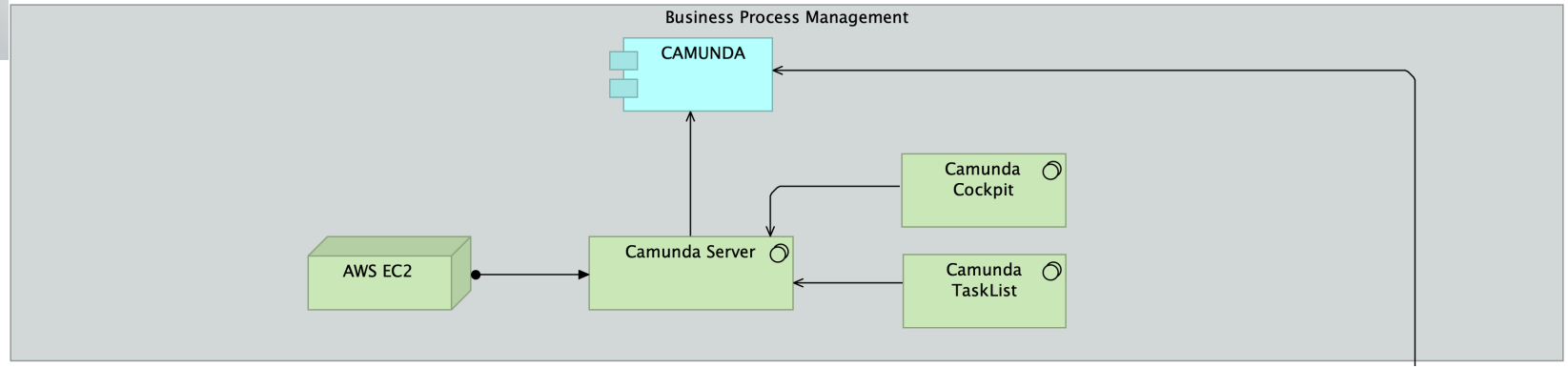
Data management layer





IE Technology Stack

API Gateways layer



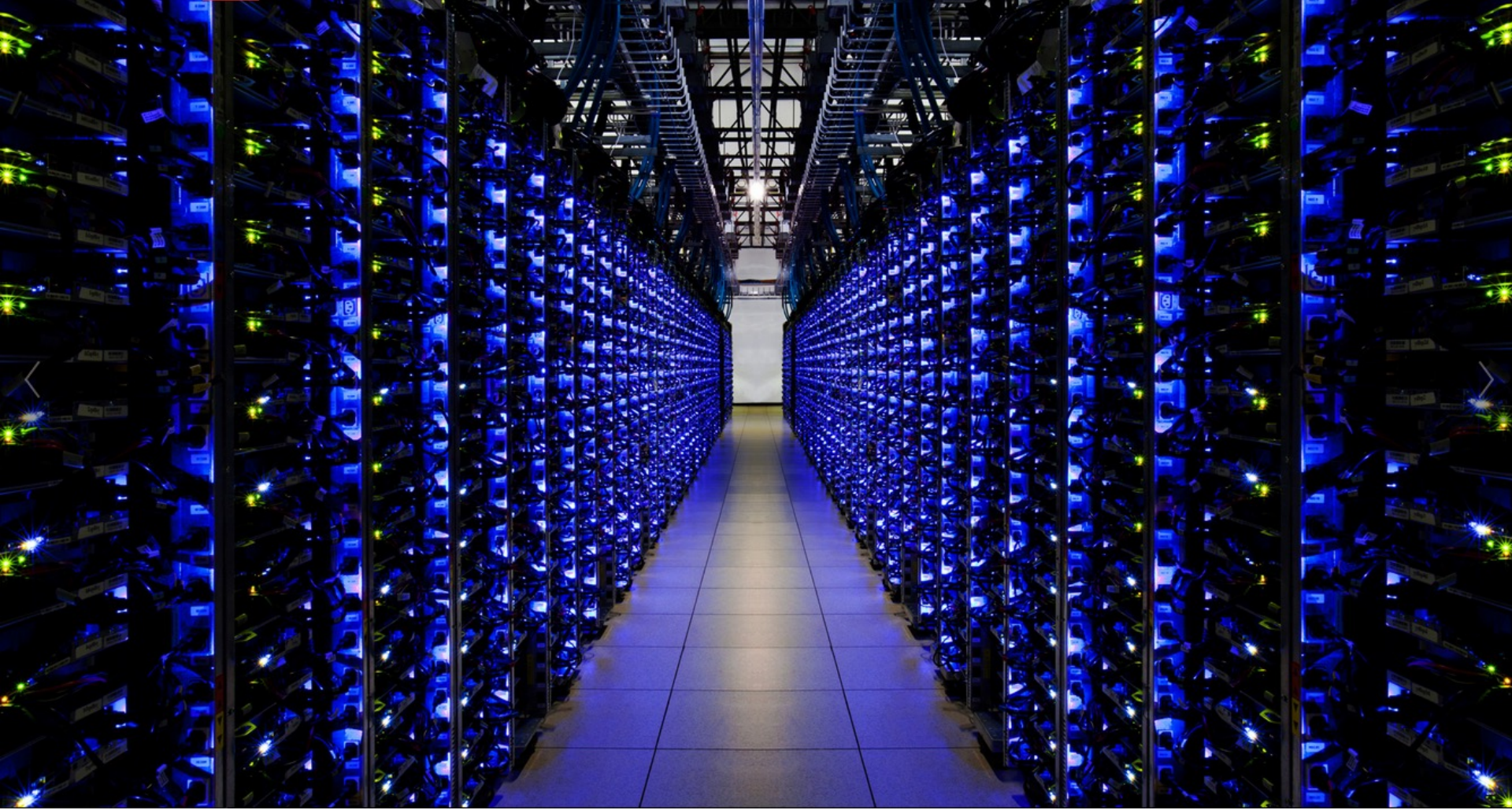
IE Technology Stack
Business Process management layer

TÉCNICO LISBOA

Digital economy

Why is enterprise integration needed?


FERRANTI
POWER
TO U.
TO V.
LOGICAL COMPUTER
ACTION SWITCHES.
OPERATION SWITCHES
OPERATION KEY
CLEAR
CLEAR ALL



From: http://www.google.com/about/datacenters/

The internet's undersea world

The vast majority of the world's communications are not carried by satellites but an altogether older technology: cables under the earth's oceans. As a ship accidentally wipes out Asia's net access, this map shows how we rely on collections of wires of less than 10cm diameter to link us all together

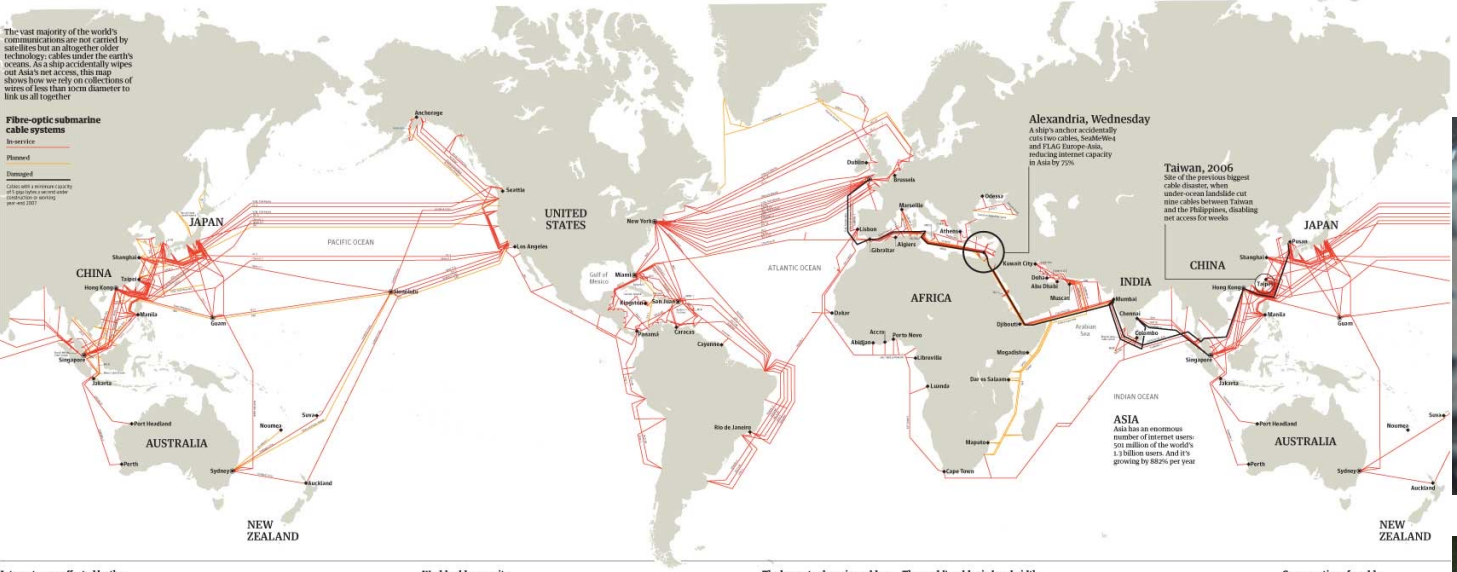
Fibre-optic submarine cable systems

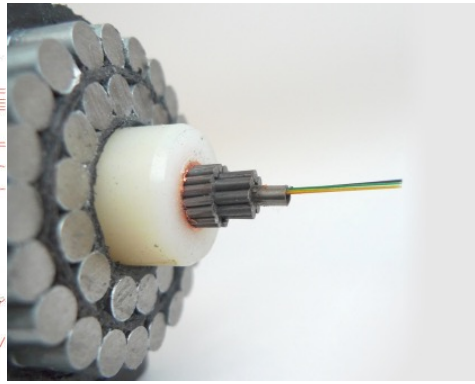
In-service

Planned

Damaged

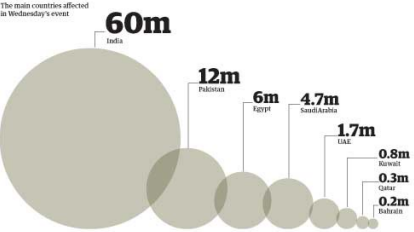
Cables with a minimum capacity of 5 tbps (bits a second) under construction or planned since mid-1997





Internet users affected by the Alexandria accident

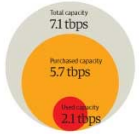
The main countries affected in Wednesday's event



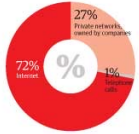
World cable capacity

Submarine cable operators light (turn on) capacity on their systems to sell bandwidth to other carriers. Carriers buy extra capacity, mainly to hold in reserve. On the trans-Atlantic route 80% of the bandwidth is purchased, but only 29% is used

Capacity in terabytes a second



What makes up "used capacity"?



The longest submarine cables

The SeaMeWe-3 system from Norden in Germany to Keoje, South Korea connects 32 different countries with 39 landing points

SeaMeWe-3	39,000 km
Southern Cross	30,500 km
China-US	30,476 km
FLAG Europe-Asia	28,000 km
South America-1	25,000 km

The world's cables in bandwidth

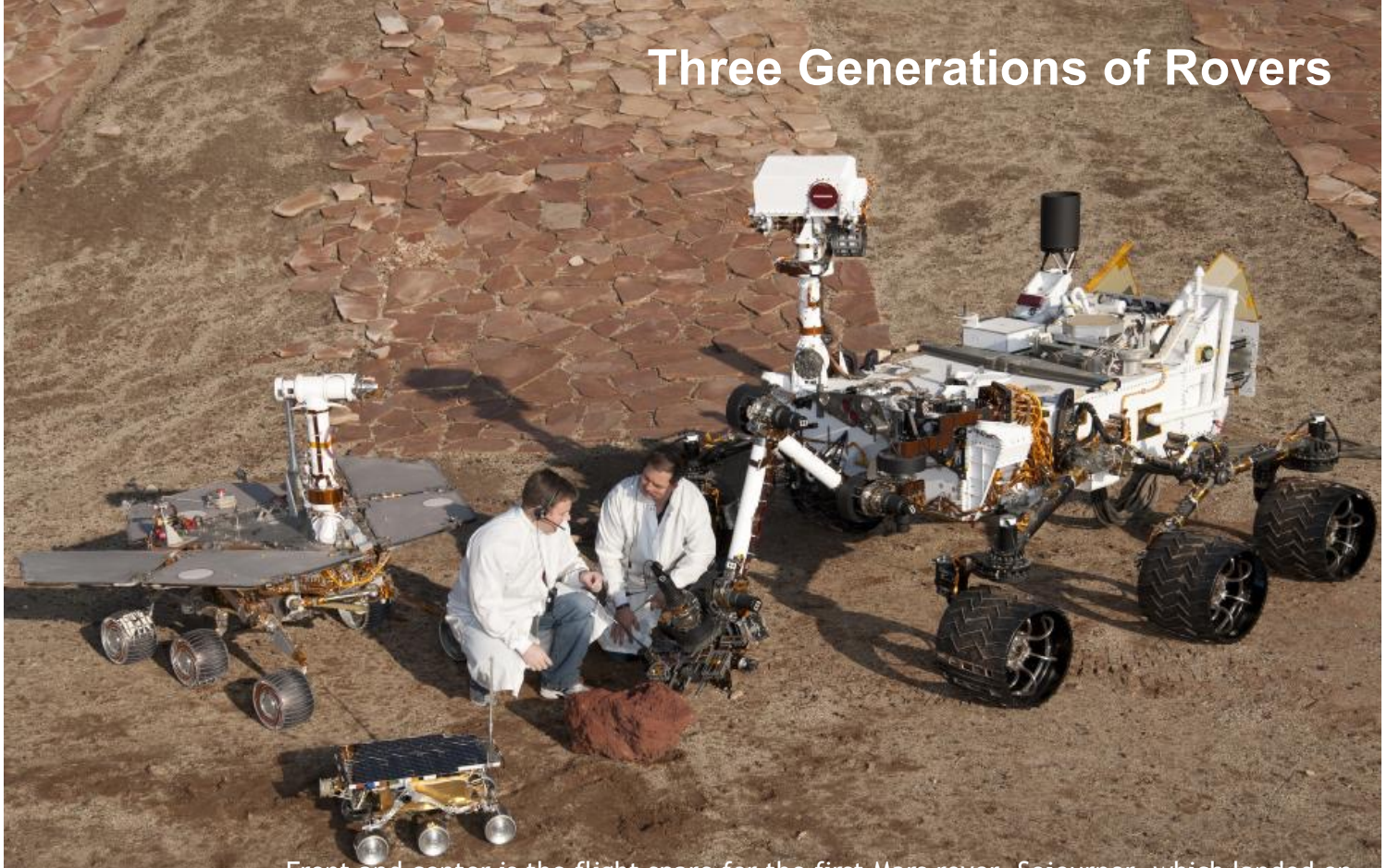
The first intercontinental telephony submarine cable system, TAT-1, connected North America to Europe in 1958 and had an initial capacity of 640,000 bytes per second. Since then, total trans-Atlantic cable capacity has soared to over 7 trillion bps

Cross-section of a cable

Cables of this strength are typically 69 mm in diameter and weigh over 10,000 kilograms a kilometer. In deeper waters, lighter and less insulated cables are used

Combining data + processing + communications

Three Generations of Rovers



Front and center is the flight spare for the first Mars rover, Sojourner, which landed on Mars in 1997 as part of the Mars Pathfinder Project. On the left is a Mars Exploration Rover Project test rover that is a working sibling to Spirit and Opportunity, which landed on Mars in 2004. On the right is a Mars Science Laboratory test rover the size of that project's Mars rover, Curiosity, which is on course for landing on Mars in August 2012.

TÉCNICO LISBOA

The Future of Jobs

TÉCNICO LISBOA

FIGURE 22 | **Top 20 job roles in increasing and decreasing demand across industries**

↗ Increasing demand		↘ Decreasing demand	
1	Data Analysts and Scientists	1	Data Entry Clerks
2	AI and Machine Learning Specialists	2	Administrative and Executive Secretaries
3	Big Data Specialists	3	Accounting, Bookkeeping and Payroll Clerks
4	Digital Marketing and Strategy Specialists	4	Accountants and Auditors
5	Process Automation Specialists	5	Assembly and Factory Workers
6	Business Development Professionals	6	Business Services and Administration Managers
7	Digital Transformation Specialists	7	Client Information and Customer Service Workers
8	Information Security Analysts	8	General and Operations Managers
9	Software and Applications Developers	9	Mechanics and Machinery Repairers
10	Internet of Things Specialists	10	Material-Recording and Stock-Keeping Clerks
11	Project Managers	11	Financial Analysts
12	Business Services and Administration Managers	12	Postal Service Clerks
13	Database and Network Professionals	13	Sales Rep., Wholesale and Manuf., Tech. and Sci.Products
14	Robotics Engineers	14	Relationship Managers
15	Strategic Advisors	15	Bank Tellers and Related Clerks
16	Management and Organization Analysts	16	Door-To-Door Sales, News and Street Vendors
17	FinTech Engineers	17	Electronics and Telecoms Installers and Repairers
18	Mechanics and Machinery Repairers	18	Human Resources Specialists
19	Organizational Development Specialists	19	Training and Development Specialists
20	Risk Management Specialists	20	Construction Laborers

Source

Future of Jobs Survey 2020, World Economic Forum.

FIGURE 23 | Emerging roles clustered into the jobs of tomorrow

Care Economy

Cloud Computing

1. Site Reliability Engineer
2. Platform Engineer
3. Cloud Engineer
3. DevOps Engineer
5. Cloud Consultant
6. DevOps Manager

Content Production

1. Social Media Assistant
2. Social Media Coordinator
3. Content Specialist
4. Content Producer
5. Content Writer
6. Creative Copywriter

Data and AI

1. Artificial Intelligence Specialist
2. Data Scientist
3. Data Engineer
4. Big Data Developer
5. Data Analyst
6. Analytics Specialist
7. Data Consultant
8. Insights Analyst
9. Business Intelligence Developer
10. Analytics Consultant

Engineering

1. Python Developer
2. Full Stack Engineer
2. Javascript Developer
4. Back End Developer
5. Frontend Engineer
5. Software Developer Dotnet
7. Development Specialist
8. Technology Analyst

Green Economy

Marketing

1. Growth Hacker
2. Growth Manager
3. Digital Marketing Specialist
4. Digital Specialist
5. Ecommerce Specialist
6. Commerce Manager
6. Head Of Digital
8. Digital Marketing Consultant
9. Digital Marketing Manager
10. Chief Marketing Officer

People and Culture

1. Information Technology Recruiter
2. Human Resources Partner
3. Talent Acquisition Specialist
4. Business Partner
5. Human Resources Business Partner

Product Development

1. Product Owner
2. Quality Assurance Tester
3. Agile Coach
4. Software Quality Assurance Engineer
5. Product Analyst
6. Quality Assurance Engineer
6. Scrum Master
8. Digital Product Manager
9. Delivery Lead

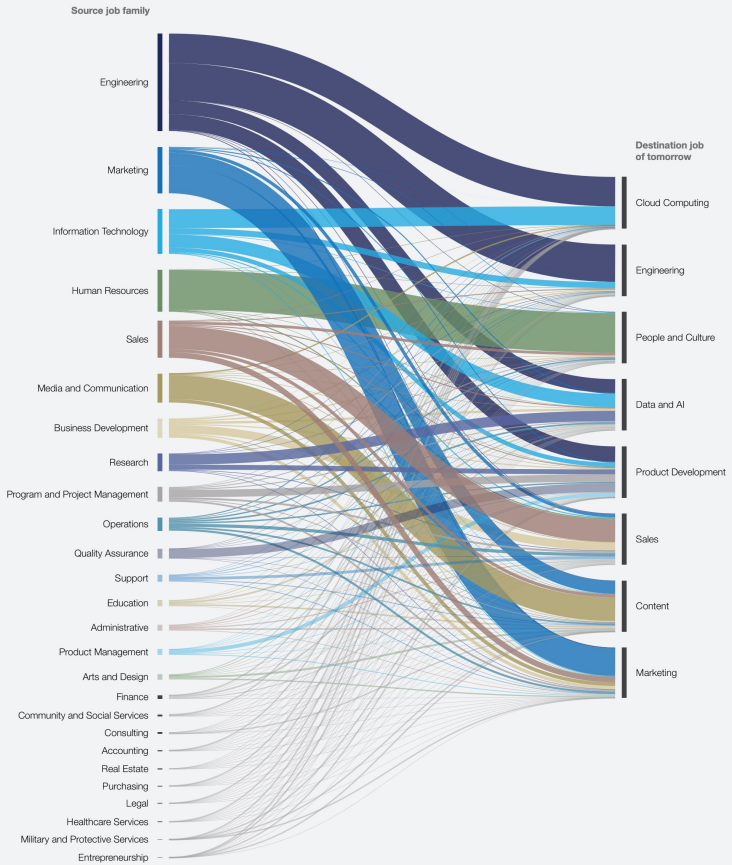
Sales

1. Customer Success Specialist
2. Sales Development Representative
3. Commercial Sales Representative
4. Business Development Representative
5. Customer Specialist
6. Partnerships Specialist
7. Chief Commercial Officer
8. Head Of Partnerships
9. Enterprise Account Executive
10. Business Development Specialist
11. Chief Strategy Officer
12. Head Of Business Development

(#) Rank Niche Mass

Source
LinkedIn Economic Graph.

FIGURE 25 | Transitions into the jobs of the future

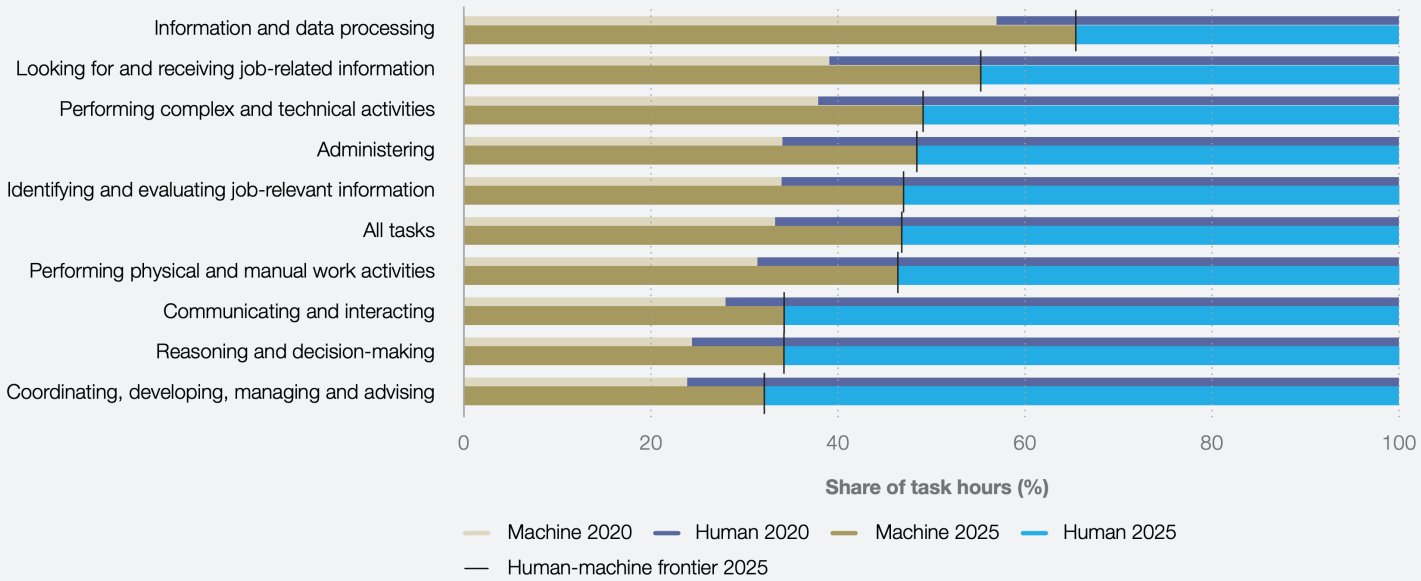


Source
LinkedIn Economic Graph.

Machines taking working hours….

FIGURE 21

Share of tasks performed by humans vs machines, 2020 and 2025 (expected), by share of companies surveyed

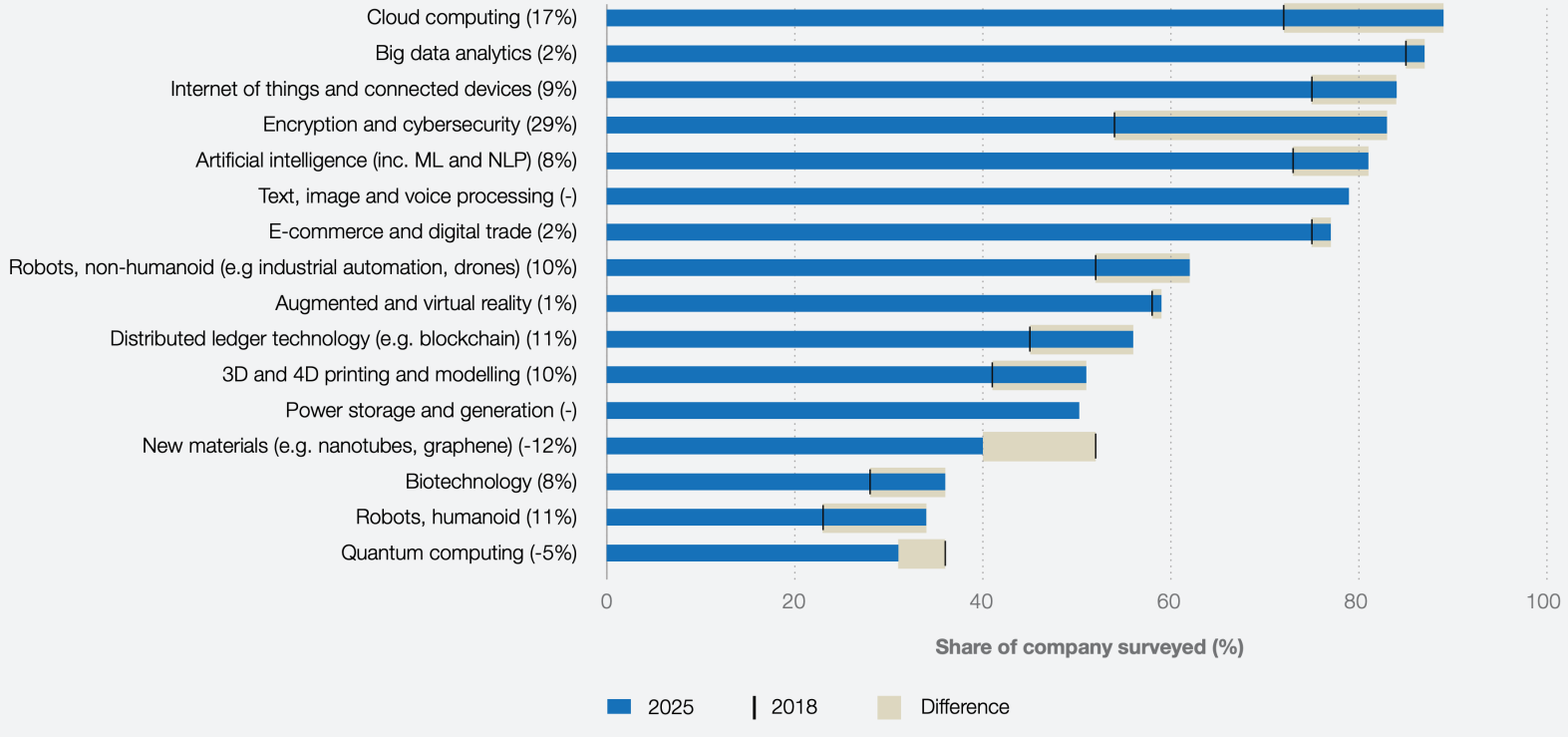


Source

Future of Jobs Survey 2020, World Economic Forum.

FIGURE 18

Technologies likely to be adopted by 2025 (by share of companies surveyed)



Source

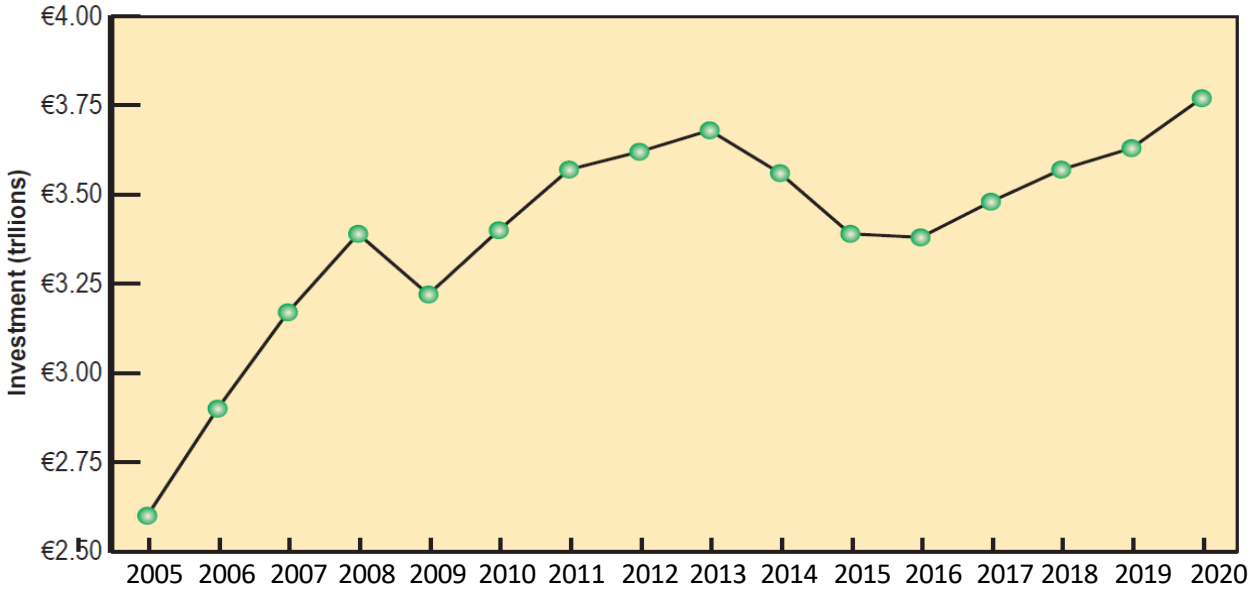
Future of Jobs Survey 2020, World Economic Forum.

Digital Transformation

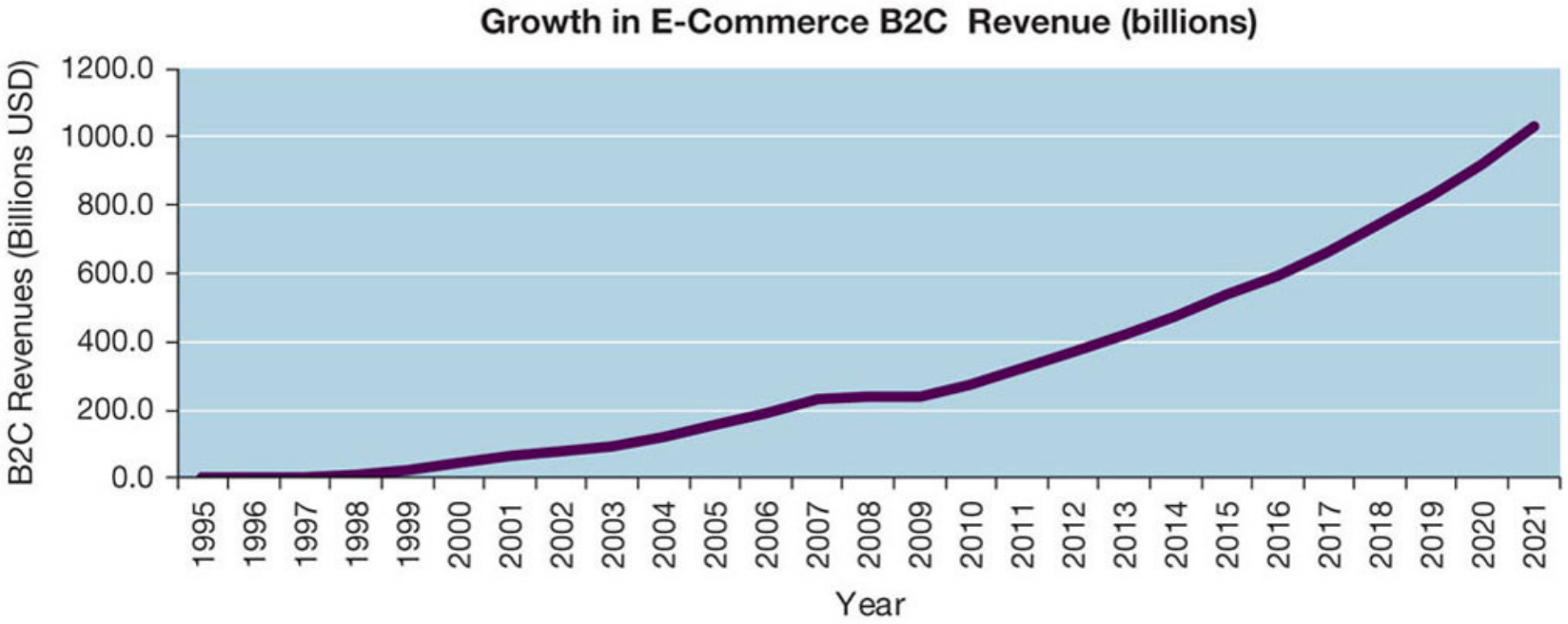
INFORMATION TECHNOLOGY CAPITAL INVESTMENT

TÉCNICO LISBOA

IT investment now accounts for an estimated 20% of all capital investment (Laudon & Laudon, 2017)



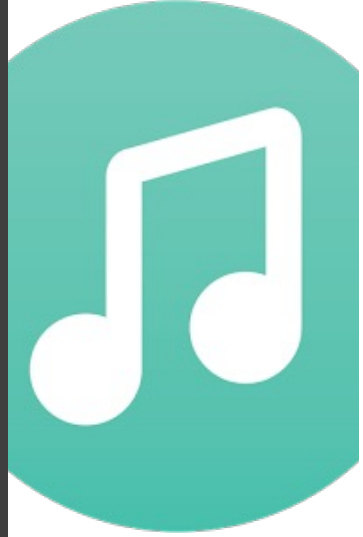
The Growth of E-Commerce

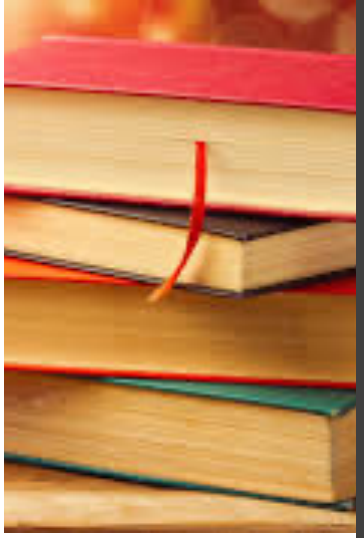


Laudon & Laudon (2020). Management Information Systems: Managing the Digital Firm, Sixteenth Edition - Global Edition, Pearson Education.

Digital landscape everywhere

Traditional goods exchanged using digital markets – *Amazon warehouse*





Goods *versus* Digital Goods

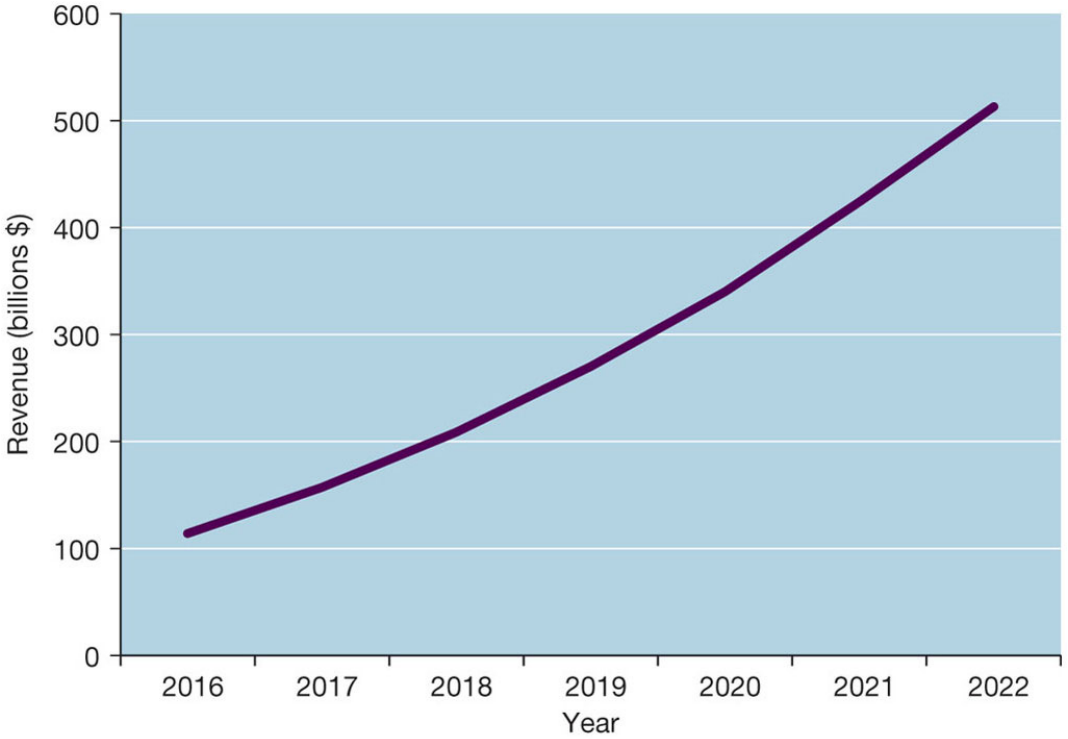
…
Books
Software
Newspapers
Magazines
……etc.

TÉCNICO LISBOA

mixing things –

digital markets using traditional environments

Mobile Retail Commerce Revenues

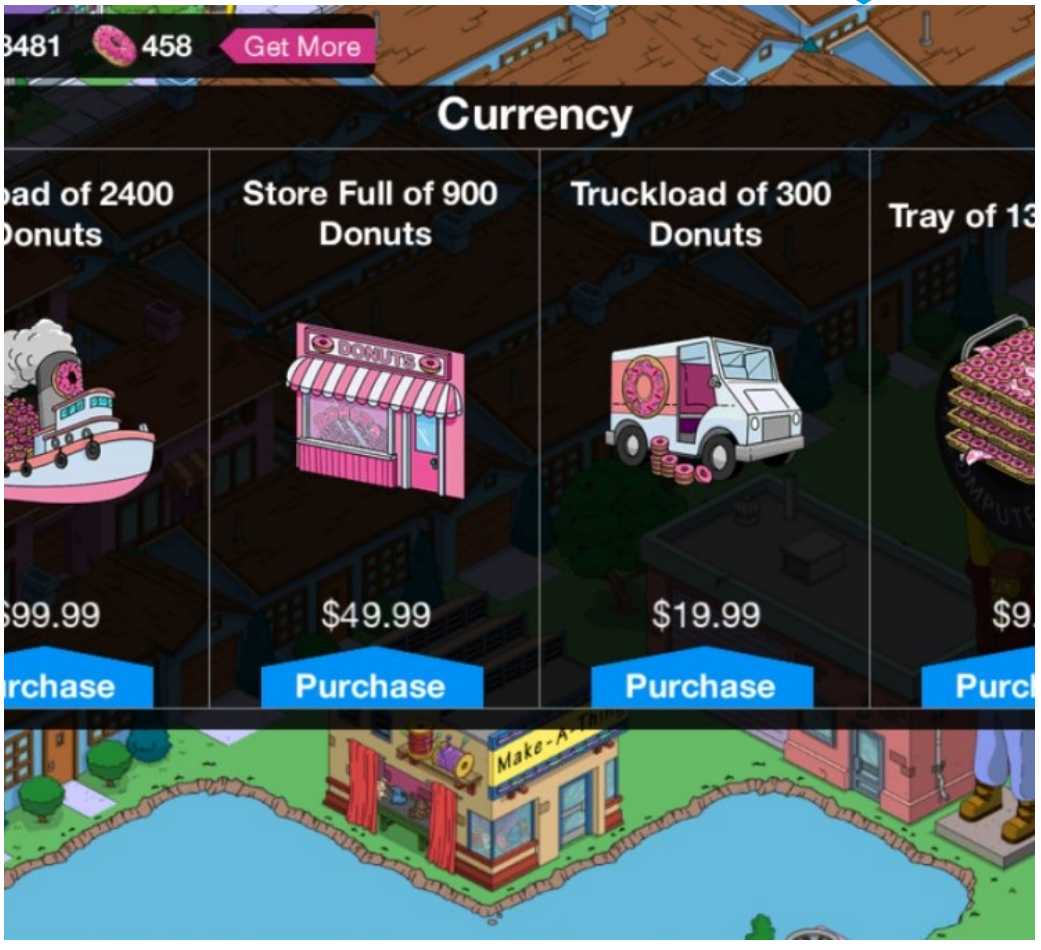


Mobile environments

Location based services and applications

Mobile environments

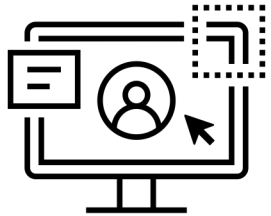
mobile games
micro-payments



TÉCNICO LISBOA

Enterprise Systems Applications

How is a digital enterprise designed?

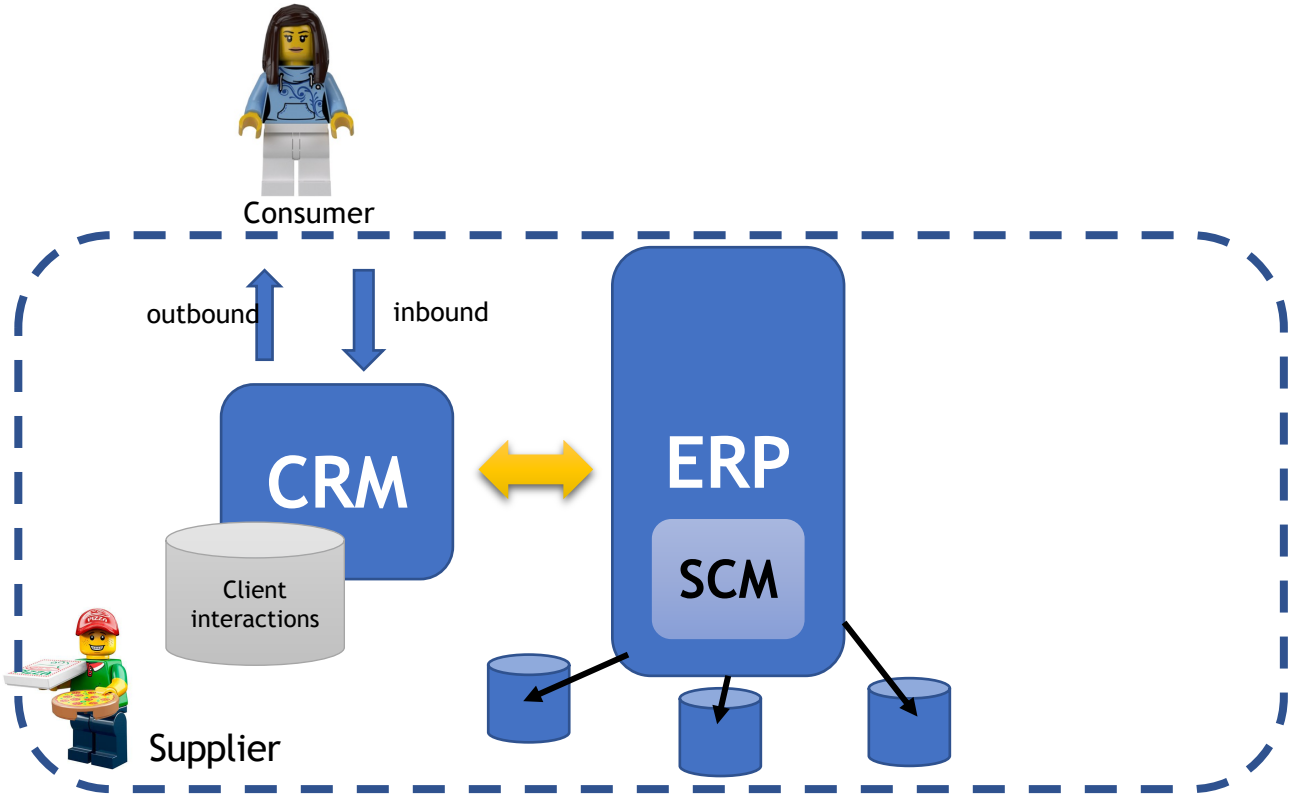


Enterprise Information Systems (EISs)

*C*an be defined as “software systems for business management, encompassing modules supporting organisational functional areas such as planning, manufacturing, sales, marketing, distribution, accounting, financial, human resources management, project management, inventory management, service and maintenance, transportation and e-business”.

They are made of computers, software, people, processes and data.

Romero, D., & Vernadat, F. (2016). Enterprise information systems state of the art: Past, present and future trends. *Computers in Industry*, *79*, 3-13.



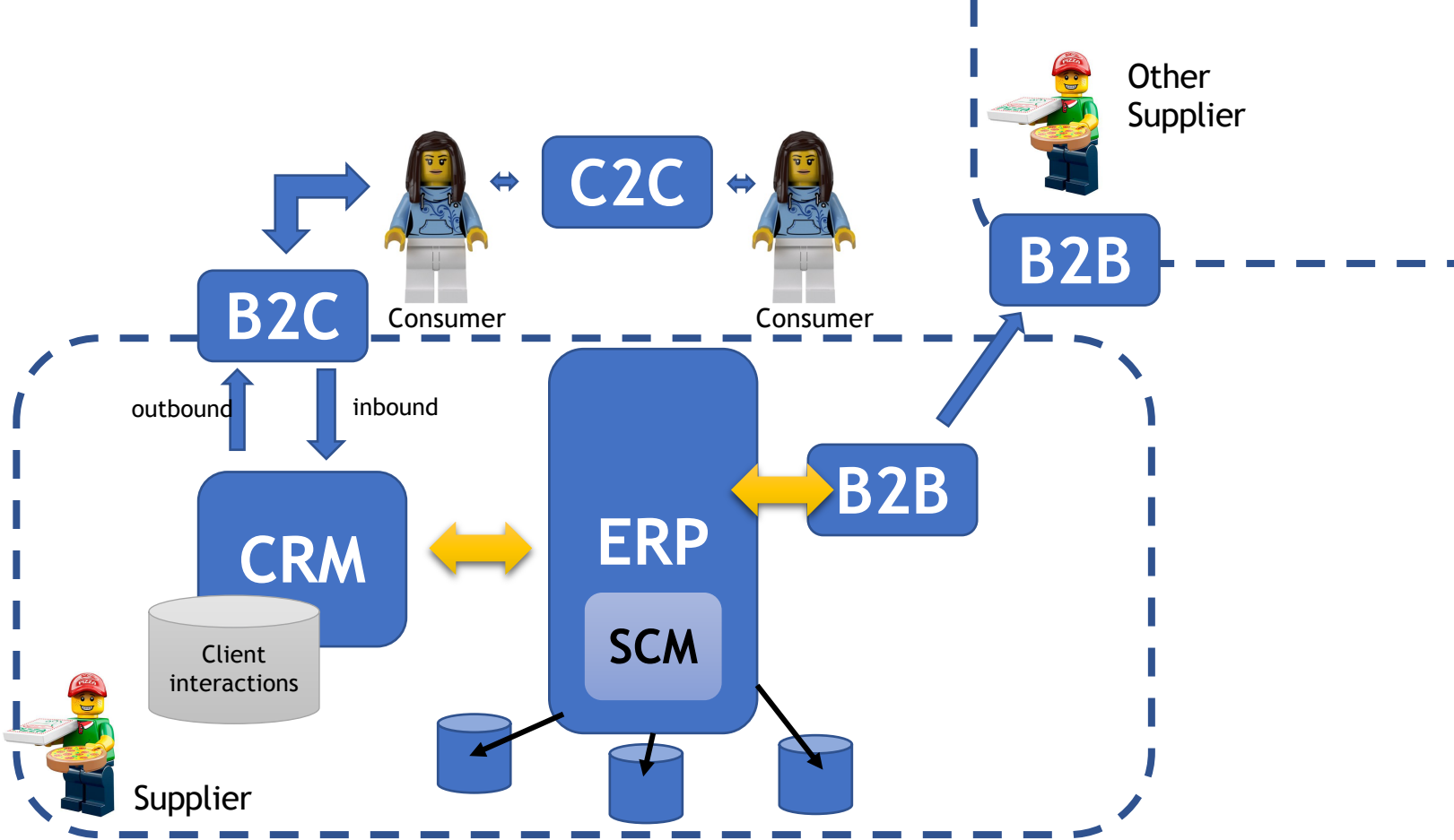
Supply Chain Operating supported by TPS (Transaction processing systems)

Customer Relationship Management Systems

- Knowing the customer
  - In large businesses, too many customers and too many ways customers interact with firm
- Customer relationship management (CRM) systems
  - Capture and integrate customer data from all over the organization
  - Consolidate and analyze customer data
  - Distribute customer information to various systems and customer touch points across enterprise
  - Provide single enterprise view of customers

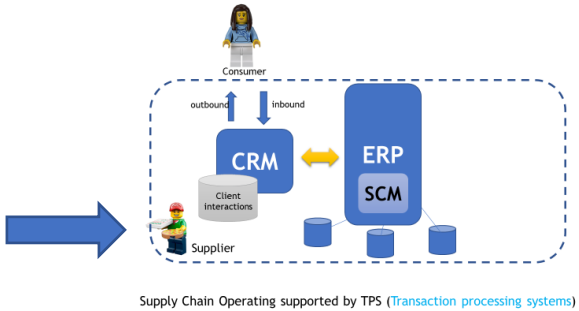
FIGURE 9.6 CUSTOMER RELATIONSHIP MANAGEMENT (CRM)

CRM systems examine customers from a multifaceted perspective. These systems use a set of integrated applications to address all aspects of the customer relationship, including customer service, sales, and marketing.



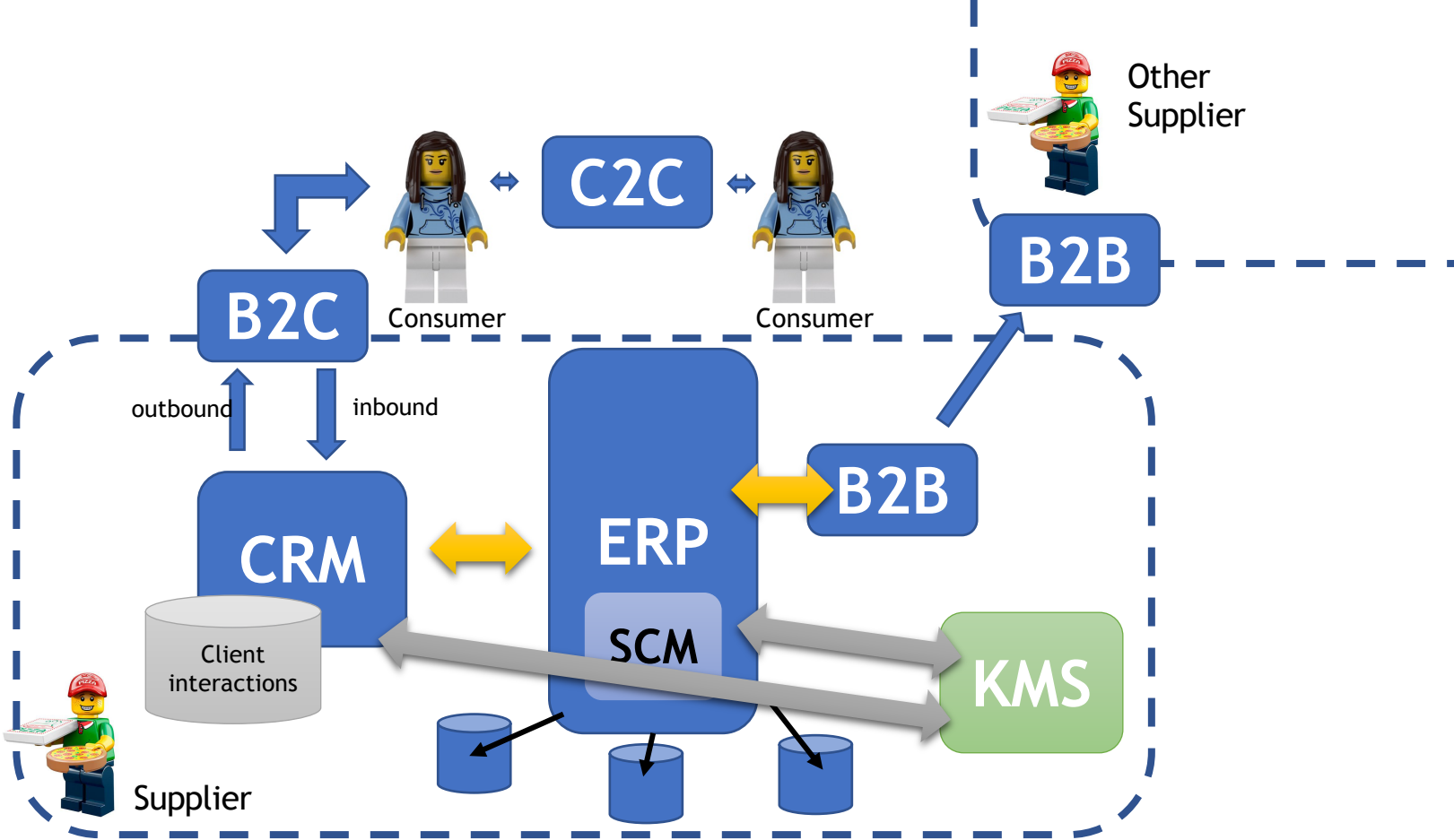
Supply Chain Operating supported by TPS (Transaction processing systems)

Supply chain management systems SCMs

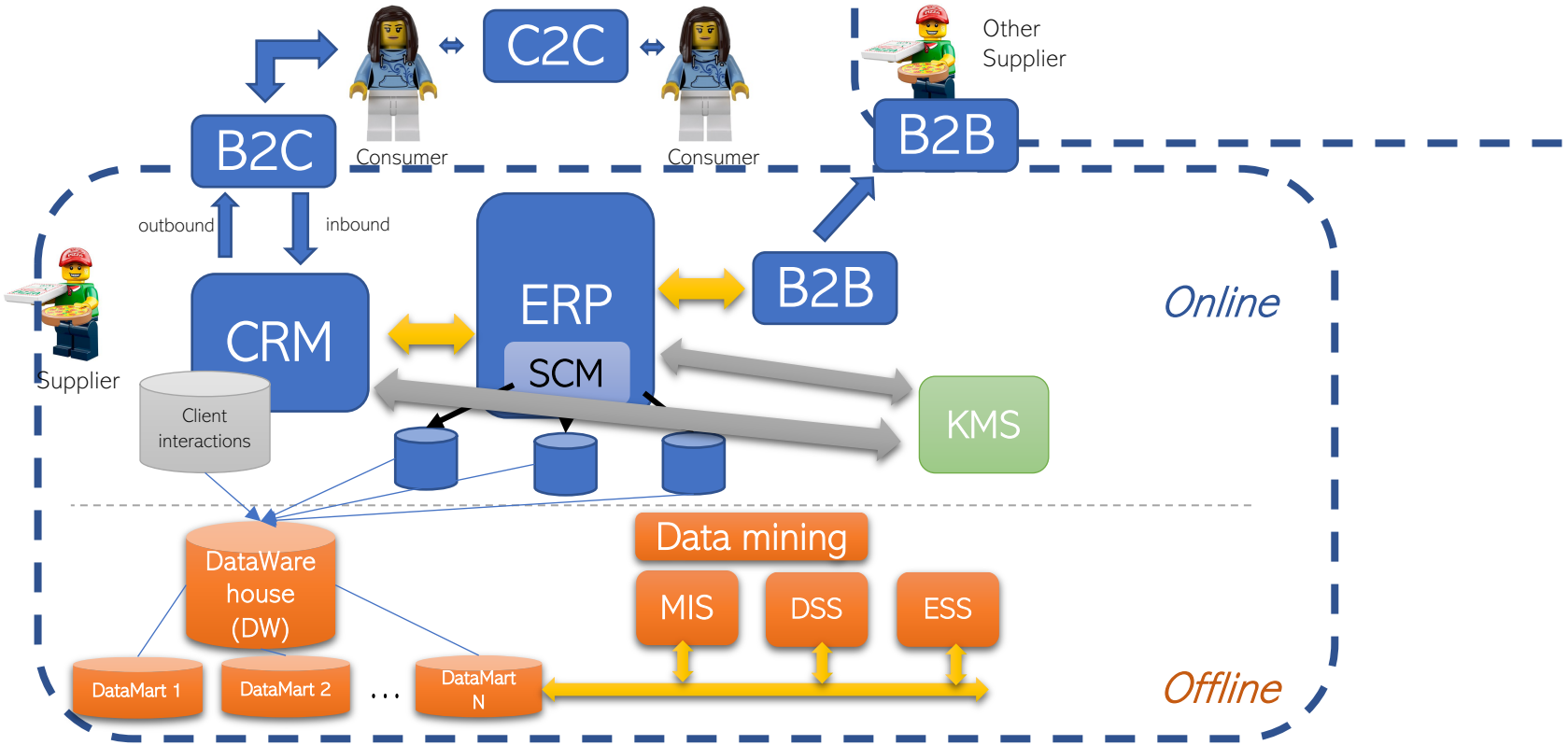


Supply Chain Operating supported by TPS (Transaction processing systems)

- Push-based model (build-to-stock)
  - Schedules based on best guesses of demand
- Pull-based model (demand-driven)
  - Customer orders trigger events in supply chain
- Sequential supply chains
  - Information and materials flow sequentially from company to company
- Concurrent supply chains
  - Information flows in many directions simultaneously among members of a supply chain network



Supply Chain Operating supported by TPS (Transaction processing systems)


C2C
Other
Supplier
B2C
Consumer
Consumer
B2B
outbound
inbound
ERP
B2B
Online
CRM
SCM
Supplier
Client
interactions
KMS
Data mining
DataWare
house
(DW)
MIS
DSS
ESS
DataMart 1
DataMart 2
...
DataMart
N
Offline

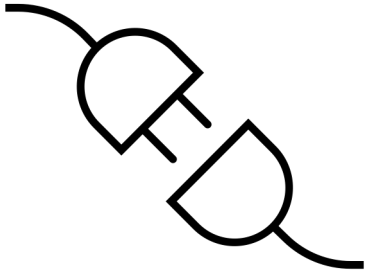
The evolution of EISs

Table 1
Enterprise information systems generations (extended from [4]).

EIS Generation	1 Application centric	2 Data centric	3 Process centric	4 Human centric	5 Things centric	6 Everything centric
Timeline	Around 80's	Around 90's	Around 00's	Around 10's	Around 15's	Around 20's
Iconic technology	MRP	ERP	BPM	HMI	Sensors	Devices
Technology drivers	Databases	Dbms, client-server architecture	Internet, soa	Semantic networks, intelligent social media, Cloud computing, Virtual and augmented reality	Smart things, wireless sensor networks, Big data, Service science, cloud computing	Real plug and play systems, open and trusted Platforms, trustworthy infrastructures, interoperability service utility
Deployment environment	Local systems		Digital systems		Cyber-physical systems	
Business challenge	Efficacy	Efficiency	Effectiveness	Resilience	Sensitiveness	Proactiveness
Organisational challenge	Support of departments	Support of enterprises	Support of supply chains	Support of social networks	Support of sensor networks	Support of interplay networks
Technology challenge	Systems integration		Systems interoperability			
Knowledge challenge	Structured data	Integrated data	Dynamic data	Real-time data		Inferred data

Panetto, H., Zdravkovic, M., Jardim-Goncalves, R., Romero, D., Cecil, J., & Mezgár, I. (2016). New perspectives for the future interoperable enterprise systems. *Computers in industry*, 79, 47-63.

TÉCNICO LISBOA



Enterprise Systems Integration & Interoperability

What is **integration**?

The integration term appears very often in the literature, *i.e.*, books, journals, conferences, newspapers…*etc.*, yet, it has evolved throughout the years, namely:

- the earliest texts referred to “**System integration**”,
- afterwards it was called "**Application Integration**”
- more recently “**Enterprise integration**” and “**Enterprise interoperability**”
- And now often described as “**Enterprise cloud integration**”

Enterprise Integration *def.*?

- Is the process of ensuring the interaction between enterprise entities necessary to achieve domain objectives

- Can be approached in various manners and at various levels, *e.g.*, *(i)* physical integration, *(ii)* application integration, *(iii)* business integration, *(iv)* through enterprise modeling, and *(v)* as methodological approach to achieve consistente enterprise-wide decision-making.

Chen, D., Doumeingts, G., & Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. *Computers in industry*, *59*(7), 647-659.

TÉCNICO LISBOA

Enterprise Systems Integration

Available knowledge base and trends

TÉCNICO LISBOA

Results for

”enterprise integration” (All Fields)

from Web of Science, Google Scholar, IEEE, ACM on 17.1.2023

Technical description

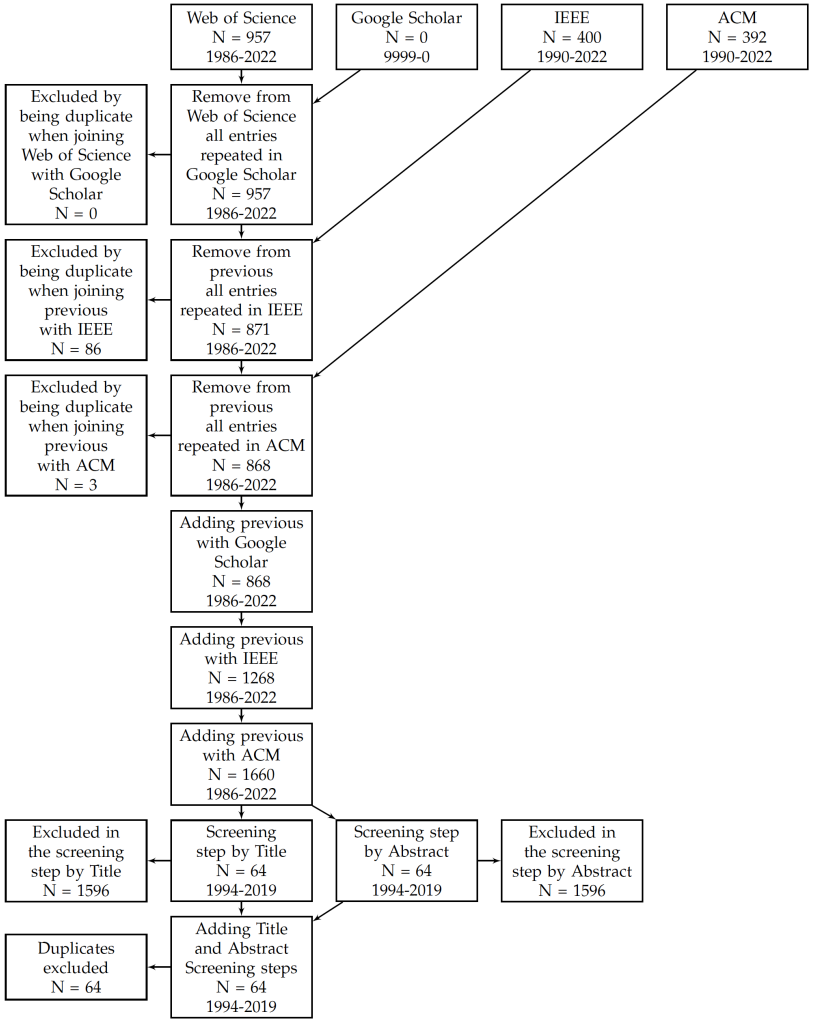


Fig. 1: The refinement steps executed in this SLR procedure and the resulting number of articles.

Results for ”enterprise integration” (All Fields)
from Web of Science, Google Scholar, IEEE, ACM on 17.1.2023

TÉCNICO LISBOA

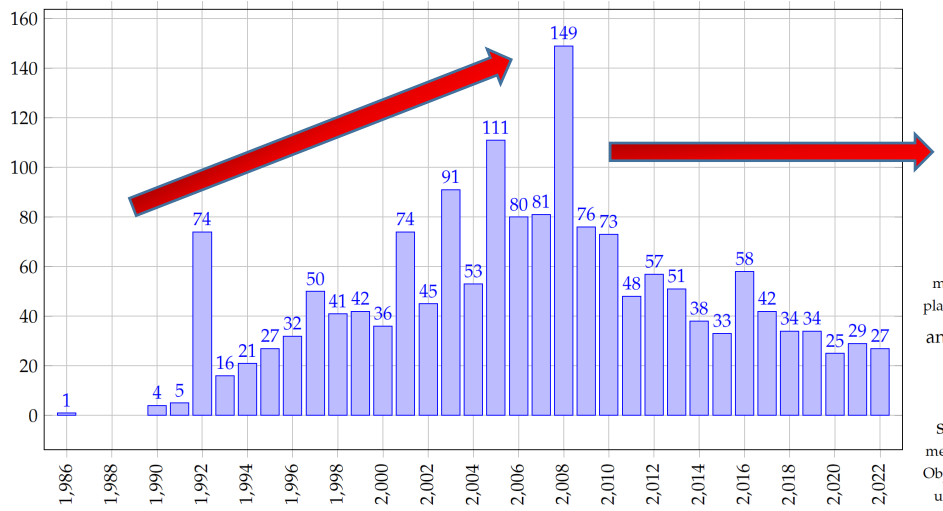
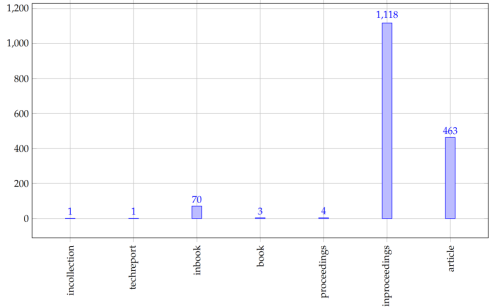


Fig. 4: Distribution of the documents by years. 1660 papers were considered.



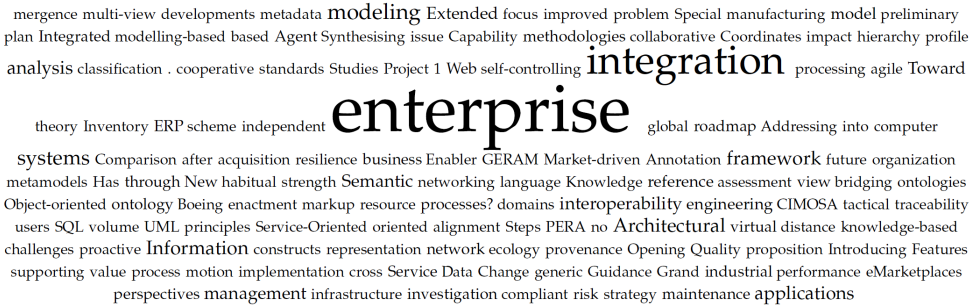


Fig. 2: Wordcloud of the main concepts extracted from Title.

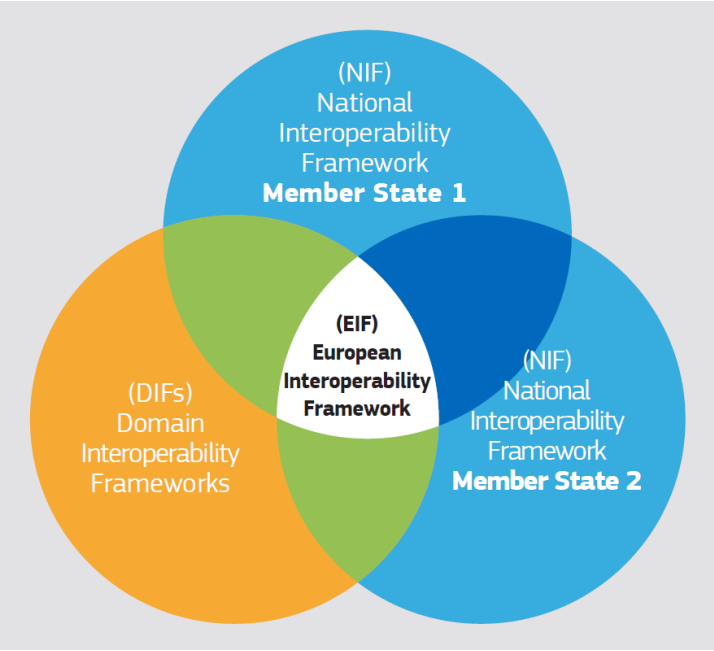
Enterprise Interoperability *def.*?

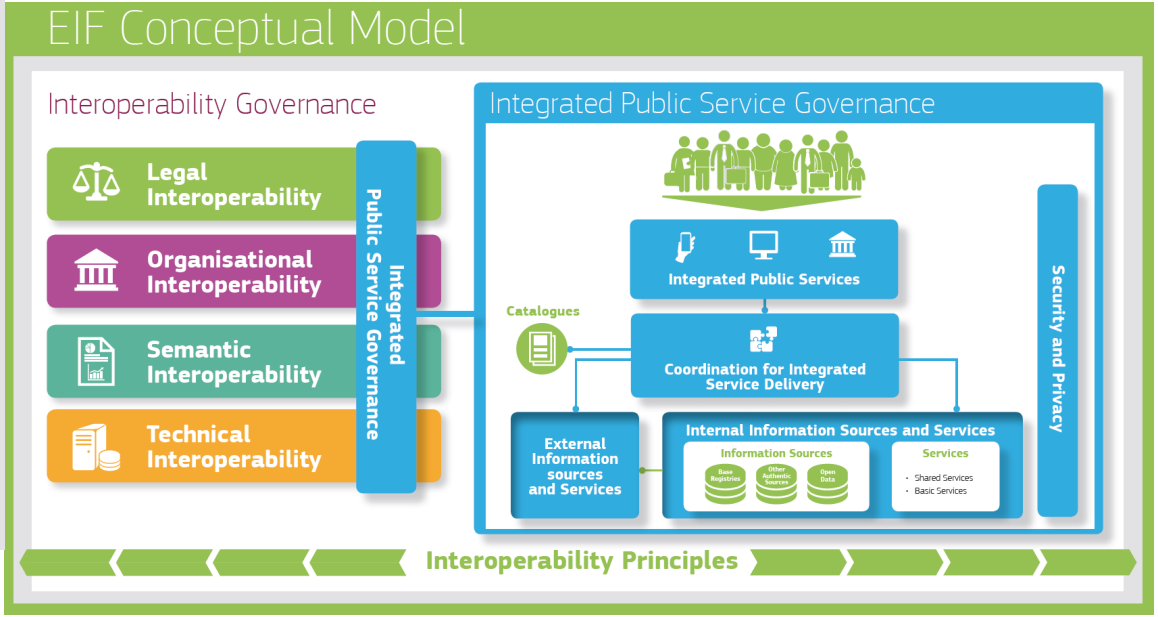
- Is the ability for two systems to understand one another and to use functionality of one another
- *"inter-operate"* implies that one system performs an operation for another system
- In the context of networked enterprises, interoperability refers to the ability of interactions (Exchange of information and services) between enterprise systems
- Interoperability is considered as significant if the interactions can take place at least on three diferente levels: **data**, **services** and **processes**, with a semantics defined in each business context

Chen, D., Doumeingts, G., & Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. *Computers in industry*, *59*(7), 647-659.

ISA² - Interoperability solutions for public administrations, businesses and citizens

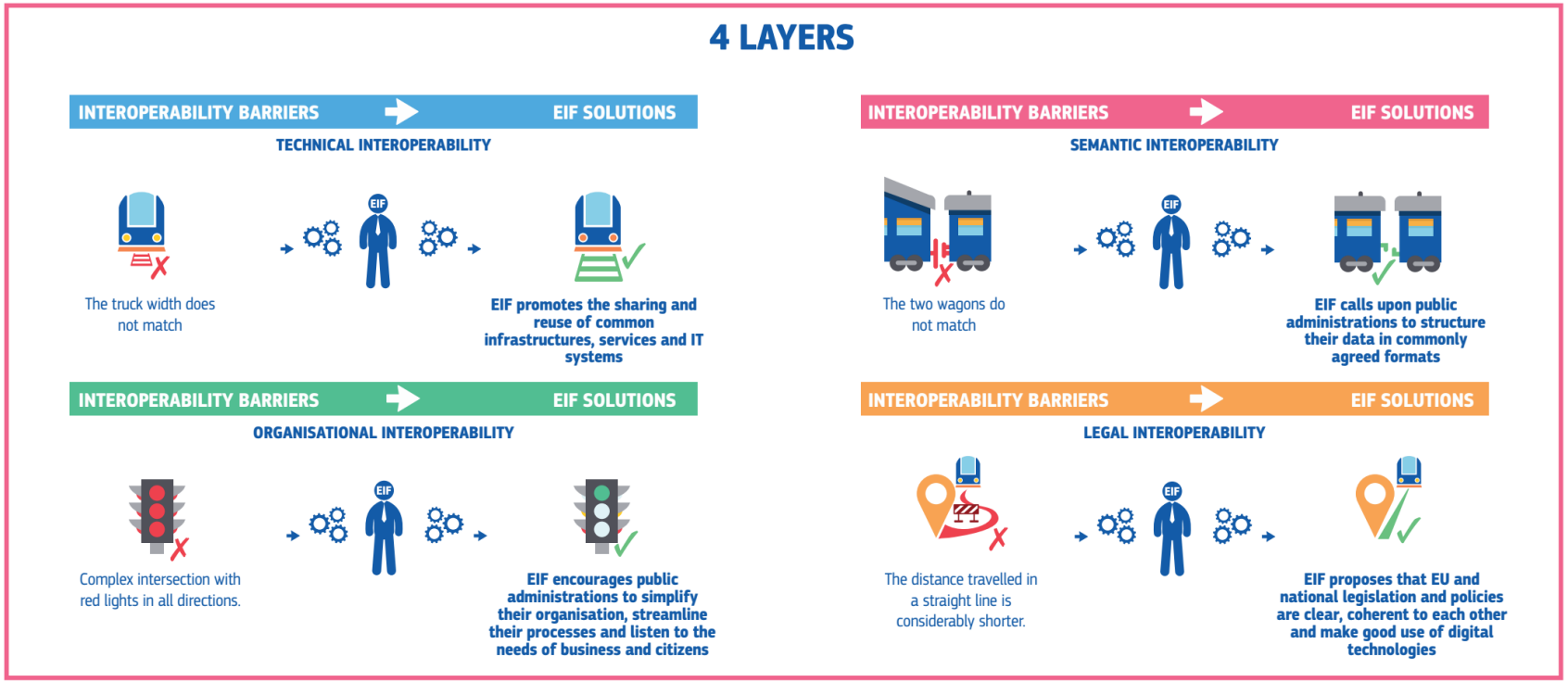
https://ec.europa.eu/isa2/eif_en/





ISA² - Interoperability solutions for public administrations, businesses and citizens

https://ec.europa.eu/isa2/eif_en/

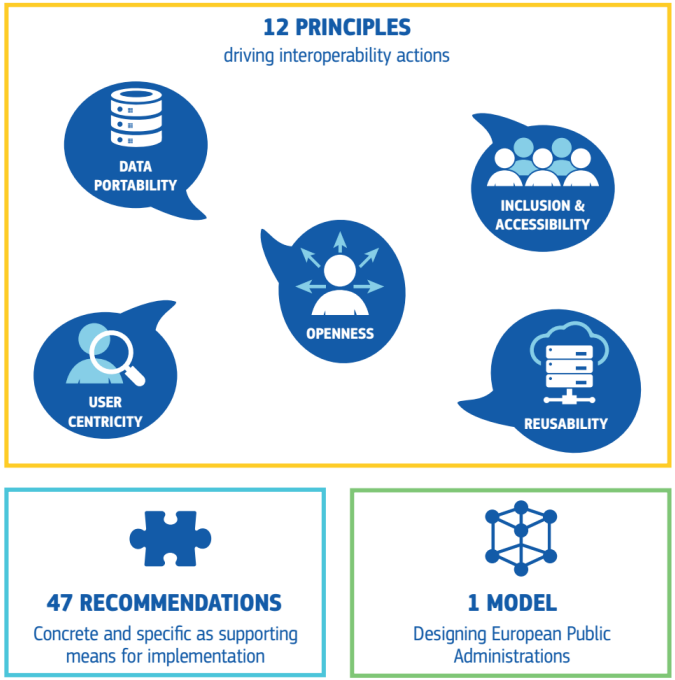


ISA² - Interoperability solutions for public administrations, businesses and citizens

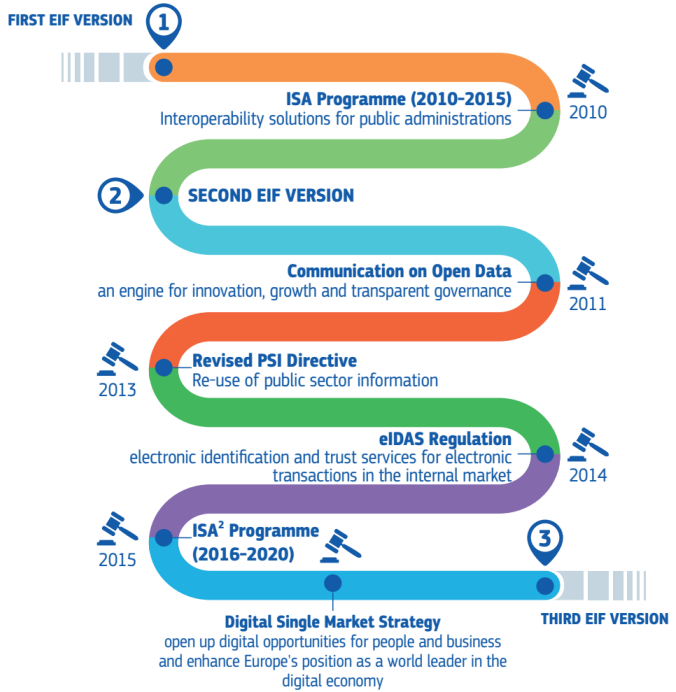
https://ec.europa.eu/isa2/eif_en/

TÉCNICO LISBOA

EIF INSIGHTS



INTEROPERABILITY STORYLINE



Integration *vs.* Interoperability?

- **Interoperability** has the meaning of coexistence, autonomy, and federated environment
  - “Loosely coupled” means that the components are connected by a communication network and can interact; they can exchange services while continuing locally their own logic of operation

- **Integration** refers more to concepts of coordination, coherence and uniformization
  - “Tightly coupled” indicates that the components are interdependent and cannot be separated

- Two integrated systems are inevitably interoperable, but two interoperable systems are not necessarily integrated

Chen, D., Doumeingts, G., & Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. *Computers in industry*, *59*(7), 647-659.

Most referred technology classes

Integrated information systems can be divided into three main classes:

(1) **Interfaced systems** representing the weakest (but still widely used) form of integration because systems can only exchange data using predefined exchange protocols and data schema (e.g. Comma-Separated Value (CSV) files over FTP (File Transfer Protocol), XML files via TCP/IP and SOAP, SQL schemas over DBlink in the case of Oracle applications, etc.),

(2) **Tightly-coupled systems** integrating all data sources by creating logical mappings between them using standardised hard-coded interfaces and predefined global schemata and requiring so-called integrating infrastructures such as Enterprise Application Integration (EAI) platforms (Linthicum, 2000) and, in between,

(3) **Loosely-coupled systems** coordinating autonomous component data sources and software applications with a set of federated schemas and open data exchange formats and protocols, preferably XML formats and using, for instance, Enterprise Service Buses (ESB) for message routing (Chappell, 2004). The latter case equates to interoperable enterprise information systems discussed in the subsequent section. Of course, there could be many intermediate gradations of IS integration between these two extrema (i.e. interfaced and tightly-coupled).

Romero, D., & Vernadat, F. (2016). Enterprise information systems state of the art: Past, present and future trends. *Computers in Industry*, *79*, 3-13.

Most referred technology for integration

File Transfer

- Flat files, structured files, XML files

Capturing User Interface – Screen scraping

- Legacy screen scraping
- HTML scraping

Synchronous messages

RPC and RMI Remote Procedures and Remote Methods

- Within local networks – J2EE, .Net

Web Protocols

- Web services for application integration
- REST – for tacital adhoc integration

Data Oriented Integration

- Strongly coupled, development effort involved

API for Package Applications

- Very efficient but requires dedicated know-how

Persistent Messages – MOM

Transaction Oriented – Transactional Monitors

File Transfer

Integration tools have typically a mechanism for transferring and transforming files with various formats

- Flat file, e.g., Comma-separated Values, is an extremely common flat-file format they're easily consumed by Google Spreadsheet, Microsoft Excel, and countless other applications
- Structured files
- XML and JSON files - file transfer universal solving practically all heterogeneity problems

- Stages of communication
  - Encoding – information (object) to file
  - File transfer
  - Decoding – file to object

File Transfer

Advantages

- All operating systems and programming languages support files
- Many applications have ways of exporting or importing files
- Easy to transfer remotely with file transfer protocols, e.g., FTP
- Support disconnected interaction

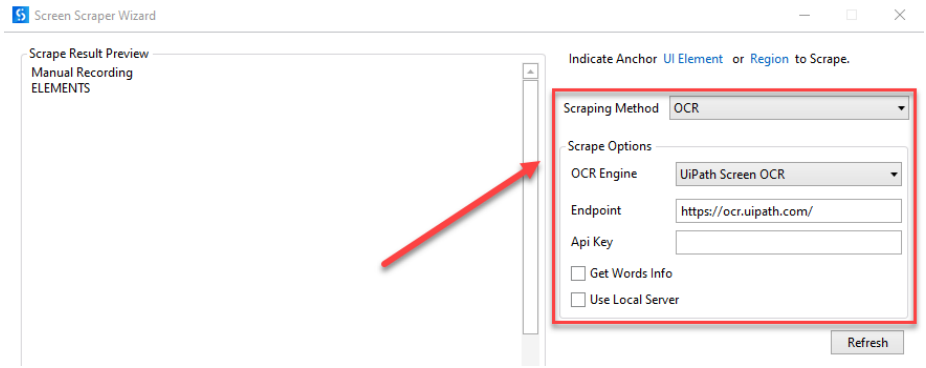
Disadvantages

- The complexity of encoding and decoding increases exponentially with complexity of the information to transfer
- Performance is limited

Capturing User Interface – Screen scraping

- Extract information directly from the user interface of an application

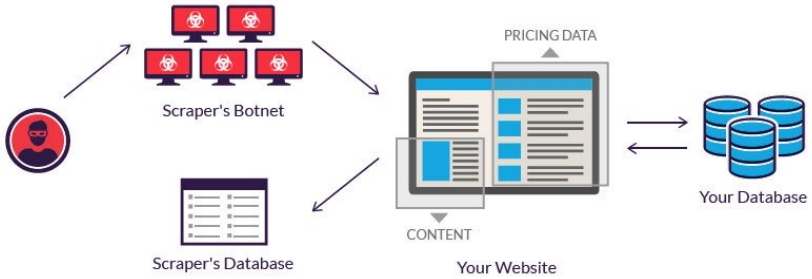
- Integration steps
  - Define the screens to use
  - Create a template indicating input and output fields
  - Replace the terminal with a system that simulates a user, sending and receiving data for each screen



UIPath

Web scraping

- Screen scrapers extract information from HTML and other markup languages
- Browsers and other web crawlers use many scraping techniques
- However, most web pages are intended for human consumption and often mix contents with presentation
- Due to widespread scraping, several anti-screen scraping techniques were developed ...



Capturing User Interface – Screen scraping

Advantages

- Suitable for integrating applications with no internal information. For example, COBOL programs on legacy mainframes
- No changes needed in the application
- No direct access to the application data

Disadvantages

- The application interfaces were not designed for integration purposes
- It is not trivial for a program to simulate a user
- The user interface may be volatile
- Performance is low
- Can be unstable due to communication problems, server availability, etc.

Synchronous messages

- In synchronous message passing, the components are processes, and processes communicate in atomic, instantaneous actions called rendezvous. **If two processes are to communicate, and one reaches the point first at which it is ready to communicate, then it stalls until the other process is ready to communicate**. “Atomic” means that the two processes are simultaneously involved in the exchange, and that the exchange is initiated and completed in a single uninterruptable step
- In the synchronous mode problems arise when **communication links or communicating threads are in an erroneous state** (broken links, threads in infinite loops etc.) and thus communicating threads remain blocked, since communication cannot be initiated or completed.

Lee, E. A. (2002). Embedded software. In *Advances in computers* (Vol. 56, pp. 55-95). Elsevier.

Efthivoulidis, G., Verentziotis, E., Meliones, A., Varvarigou, T., Kontizas, A., & Deconinck, G. (1999). Fault Tolerant Communication in Embedded Parallel High Performance Computing. In Parallel Computational Fluid Dynamics 1998 (pp. 405-414). North-Holland.

RPC and RMI Remote Procedures and Remote Methods

- **Remote Procedure Calls (RPCs)** were developed by Sun Microsystems in the early 1980s. RPCs are represented on different operating systems, including most Unix and MS Windows systems. "Windows NT, for example, supports lightweight RPCs across processes and, with DCOM, full RPCs." (see [15], p.172)
- **RPC middleware** enables one application **to trigger a procedure** in another application - **running on the same computer or on a different computer or network** - as if both were part of the same application on the same computer.

- **Advantages**
  - RPC has a good heterogeneity support, because "RPC has bindings for multiple operating systems and programming languages."
  - Marshalling and unmarshalling are automatically generated, thus simplifying the development
- **Disadvantages**
  - RPCs don't support group communication
  - They have no direct support for asynchronous communication, replication and load balancing, therefore leading to a limited scalability
  - Fault tolerance is worse than by other middleware types, because "many possible faults have to be caught and dealt with in the program."

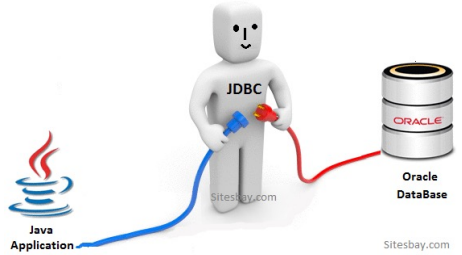
Pinus, H. (2004, January). Middleware: Past and present a comparison. In *Seminar Paper, Darmstadt University of Technology (GER), Software Technology Group.*

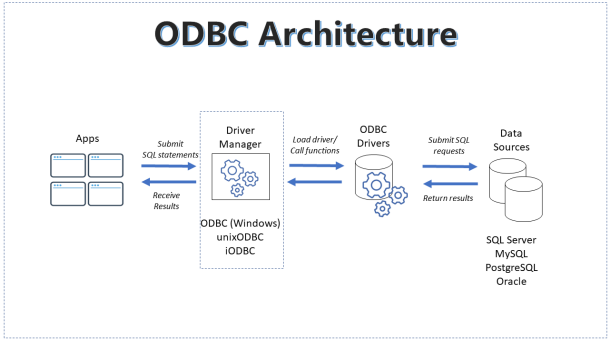
Web Protocols

- Many of the released web protocols were used throughout the years to integrate systems. It consists in well-documented connectivity, yet the complexity depend on the details of each protocol
- Web services for application integration, e.g.:
  - **Simple Object Access Protocol** (SOAP) **-** is a standard protocol originally designed to enable communication between applications developed in different languages and platforms.
  - **Representational State Transfer** (REST) - A stateless client/server protocol: each HTTP message contains all the information needed to understand the request. As a result, neither the client nor the server need to record any state of inter-message communications. In practice, many HTTP-based applications use cookies and other mechanisms to maintain session state (some of these practices, such as URL rewriting, are not allowed by the REST rule). A well-defined set of operations that apply to all information resources: HTTP itself defines a small set of operations, the most important of which are POST, GET, PUT and DELETE. Often these operations are combined with CRUD operations for data persistence, where POST does not exactly fit this scheme.

Data Oriented Integration

- ODBC, JDBC – Independent API for database management systems
- Strongly coupled, development effort always involved
- An application is independent of the DBMS if it doesn't use specific aspects such as stored procedures, triggers, or SQL specific Commands
- Remote processing imposes a performance penalty due to the way parsing and execution commands are made.





Data Oriented Integration

- **Object request broker (ORB) middleware** acts as broker between a request from one application object or component**,** and the fulfilment of that request by another object or component on the distributed network.

- ORBs operate with the **Common Object Request Broker Architecture (CORBA)**, which enables one software component to make a request of another without knowing where other is hosted, or what its UI looks like - the "brokering" handles this information during the exchange.

Data Oriented Integration

Advantages

- Simple both in Microsoft and Java platforms with ODBC and/or JDBC
- Relatively low cost because it doesn't require rewriting applications
- Most SGDB manufacturers provide drivers

Disadvantages

- A large organization may have hundreds of data bases making it difficult to create an architecture
- The Data Schema must be known
- Requires to have technical knowledge on database repositories because the operation can have serious consequences on the information
- Data types can be different and there is need to transform them
- The data is not validated by the application
- A strong coupled integration – any changes affect the integration
- Replicated data can become inconsistent

API for Package Applications

API (application programming interface) middleware provides tools developers can use to create, expose and manage APIs for their applications so that other developers can connect to them.

Some API middleware includes tools for *monetizing* APIs - enabling other organizations to use them, at cost.

Examples of API middleware include API management platforms, API gateways and API developer portals.

Message-oriented middleware (MOM)

- There are two different types of MOM: message queuing and message passing.
  - Message queuing is defined as indirect communication model, where communication happens via a queue. Message from one program is sent to a specific queue, identified by name. After the message is stored in queue, it will be sent to a receiver.
  - In message passing - a direct communication model - the information is sent to the interested parties. One favour of message passing is publish-subscribe (pub/sub) middleware model. In pub/sub clients have the ability to subscribe to the interested subjects. After subscribing, the client will receive any message corresponding to a subscribed topic

- **MOM** enables application components using different messaging protocols to communicate to exchange messages. In addition to translating - or transforming - messages between applications, MOM manages routing of the messages so they always get to the proper components in the proper order.

- The publish/subscribe MOM works slightly differently. This MOM is an event-driven process. If a client wants to participate, it first joins an information bus. Then depending on its function as the publisher, subscriber, or both, it registers an event listener in the bus. The publisher sends a notice of an event to the bus (on the MOM server). The MOM server then sends out an announcement to the registered subscriber(s) that data is available. When the subscriber requests from a specific publisher some data, the request is wrapped in a message and sent to the bus. The bus then sends an event to the publisher requesting the data

Pinus, H. (2004, January). Middleware: Past and present a comparison. In *Seminar Paper, Darmstadt University of Technology (GER), Software Technology Group*.

Transaction Oriented *or* Transactional middleware (TM)

- TM, or even, transaction processing (TP) monitors were designed in order to support distributed synchronous transactions. The main function of a TP monitor is a **coordination of requests between clients and servers that can process these requests**. Request is a *"message that asks the system to execute a transaction."*

- **TM** provides services to support the execution of data transactions across a distributed network. The best-known transactional middleware are transaction processing monitors (TPMs), which ensure that transactions proceed from one step to the next - executing the data exchange, adding/changing/deleting data where needed, etc. - through to completion.

Pinus, H. (2004, January). Middleware: Past and present a comparison. In *Seminar Paper, Darmstadt University of Technology (GER), Software Technology Group.*

What is Middleware *def.*?

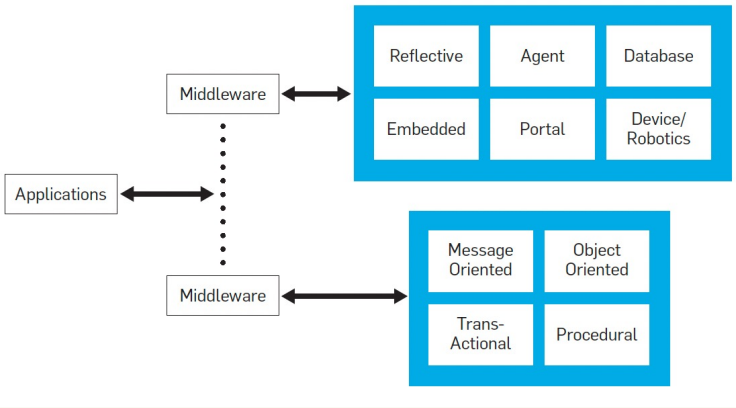
On the one hand, both a software and a DevOps engineer would describe middleware as the layer that *"glues"* together software by different system components; on the other hand, a network engineer would state that middleware is the *fault-tolerant and error-checking* integration of network connections.

In other words, they would define middleware as **communication management software**.

A data engineer, meanwhile, would view middleware as the technology responsible for **coordinating, triggering, and orchestrating actions** to process and publish data from various sources, harnessing big data and the IoT.

Given that there is no uniform definition of middleware, it is best to adopt a field-specific approach.

Figure 1. Types of middleware.



Gazis, A., & Katsiri, E. (2022). Middleware 101. *Communications of the ACM*, *65*(9), 38-42.

Middleware – the future

- This middleware "*is an approach for developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API.*"

- **This means that middleware should not serve solely as an object-oriented solution to execute simple request-response commands.**

But, middleware can incorporate **pull-push events** and **streams** via multiple **gateways** by combining **microservices** architectures to develop a holistic decentralized ecosystem**.**

Gazis, A., & Katsiri, E. (2022). Middleware 101. *Communications of the ACM*, *65*(9), 38-42.

Middleware is STILL Everywhere!!!

Middleware! It was somewhere, and now it is definitely everywhere. Over the past four decades, the term middleware has been tossed around, picked up, and investigated vigorously. **Yet, if you ask 10 different people "what is middleware?" You will most likely get 10 different answers.**

It started as some additions on top of operating systems to **facilitate complex applications development**, moved to become **data integration features**, then became **network applications facilitator**, and eventually became an **important component of every distributed environment, application, system and platform** there is.

To-date, if you examine any type of distributed system or application, you must find middleware or some middleware functionality involved. Although recently, the term itself is becoming less used, yet it still exists in **mobile and sensor networks, service-oriented architectures, grid computing, cloud computing, online multi-player games, networked robotics, the Internet of things**, and much more.

So as is, middleware is really still everywhere and most likely will remain everywhere for a very long time.

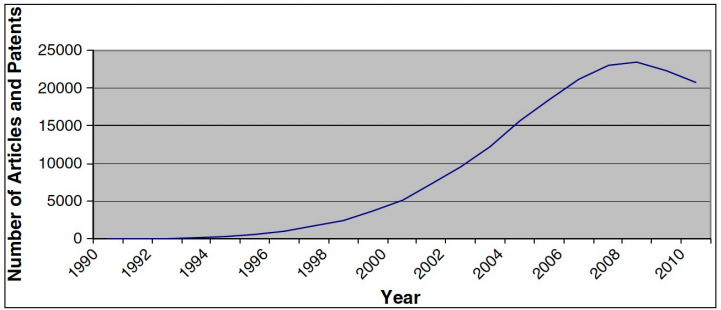


Figure 1. Appearance of "middleware" word on Google Scholar.

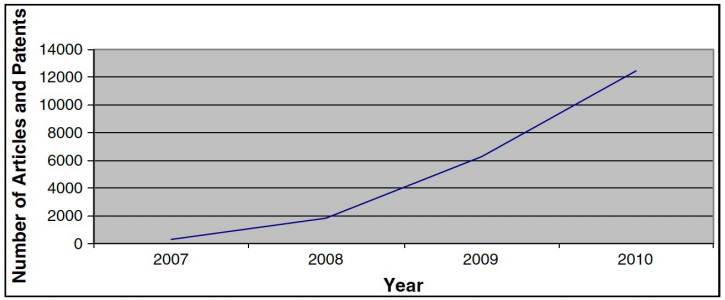


Figure 2. Appearance of "cloud computing" in Google Scholar.

Al-Jaroodi, J., & Mohamed, N. (2012). Middleware is STILL everywhere!!!. *Concurrency and Computation: Practice and Experience*, *24*(16), 1919-1926.

What is Service Oriented Architecture (SOA)?

- The core unit is **the service** that encapsulates a business function, and services are explicitly defined with interfaces that are independent of implementation (**promote the reuse**)
- Services are **loosely coupled** and invoked through communication protocols that are independent of the location
- Each service is instantiated in a **single site** and invoked remotely on this site by all applications that use it (no replicas with potential independent developments)
- There is **no inheritance** or strong dependencies between services
- Each service is created (build) once but can be deployed to all systems that require it.

https://aws.amazon.com/what-is/service-oriented-architecture/

Results for ALL=” service oriented architecture” from Web of Science on 8.3.2023

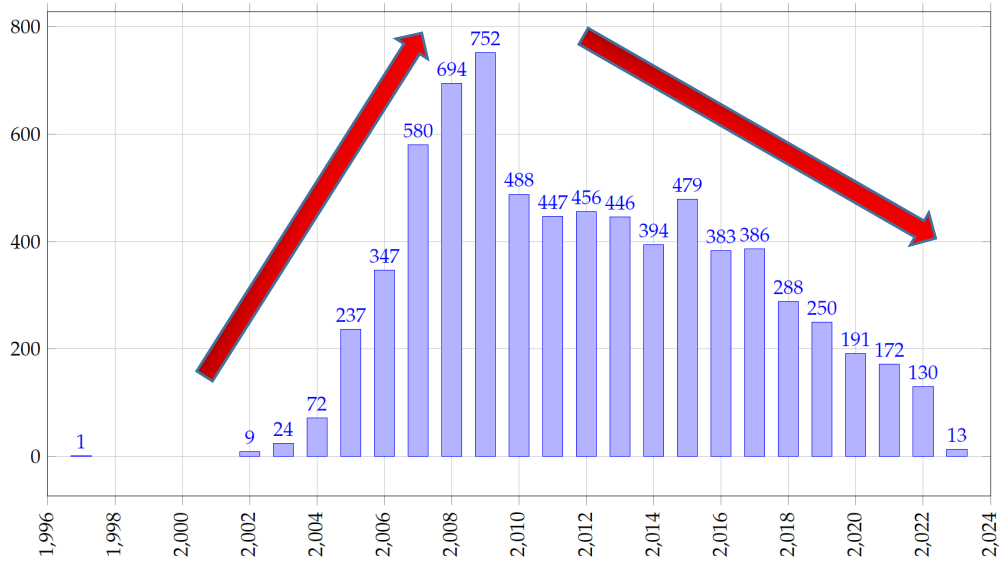
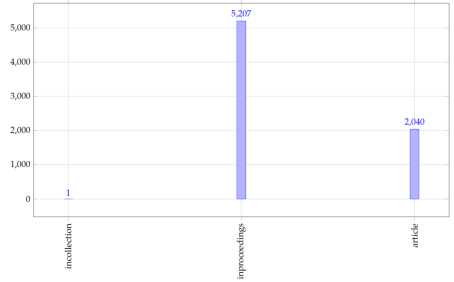
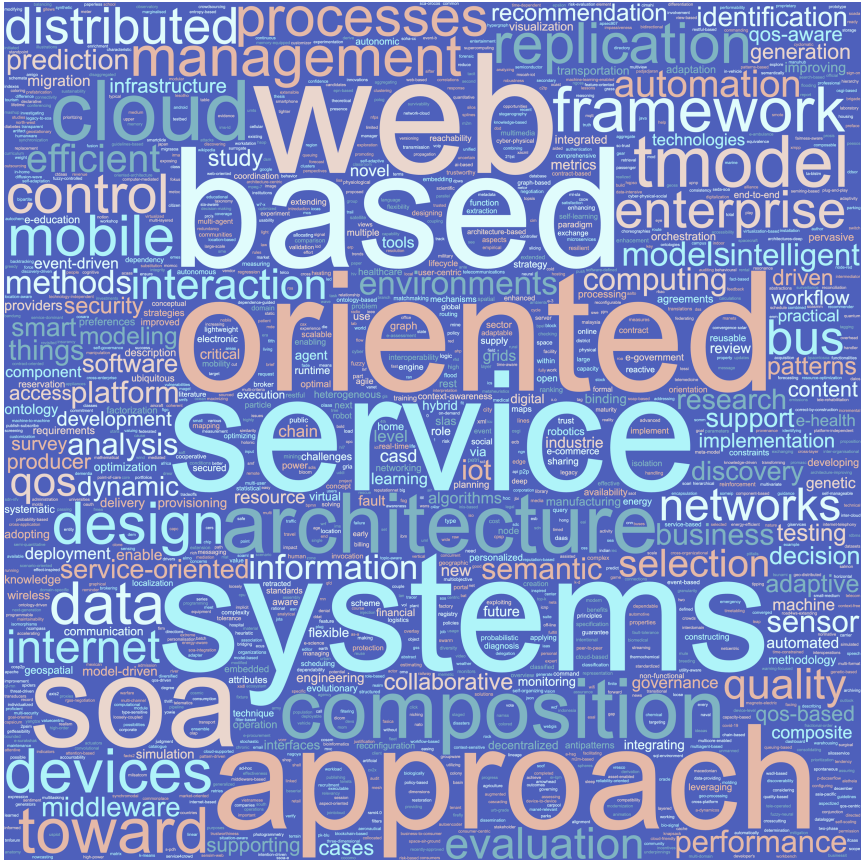
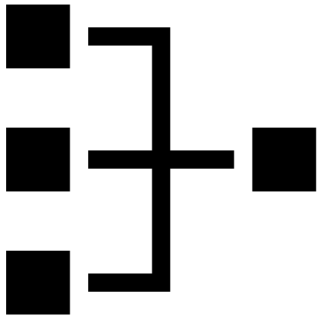


Fig. 3: Distribution of the documents by years. 7248 papers were considered.





TÉCNICO LISBOA



Enterprise Service Bus

Available knowledge base and trends

ESB

- An ESB is a system for interoperating different application systems in a service-oriented architecture (SOA). It represents a distributed computing architecture and is a variant of the client-server model in which an application can assume both a client and a server role.
- The fundamental concept of an ESB is the fact that a communication “bus” is used and placed between the different applications. In this way, applications are decoupled from each other, operating in such a way that they do not need to be aware of the other systems present on the service bus. The complexity of transport services and protocols are abstracted by the service bus, offering applications an easy way to interoperate with each other.
- The ESB concept was born out of the need to escape the point-to-point communication model (applications that interact directly with each other) that proved difficult to manage or monitor when the organization's infrastructure reaches a certain size.
- They also allow accelerating the integration between heterogeneous applications and thus respond to market needs more quickly.
- The mechanisms provided by an ESB platform are the following: connectivity, transactional control, security, data transformation, transmutations, catalog of exposed services, service orchestration, routing.
- **There are two categories for classifying ESBs: (A) open source and (B) commercial ones. Where the criteria of choice usually used to evaluate the ESB are (i) usability, (ii) the support provided, (iii) the functionality provided, (iv) possible adaptability over the functionality provided, (v) the capacity for future expandability , (vi) existing maintenance mechanisms, (vii) community size, (viii) connectivity offered, (ix) cost, and (x) licensing model.**
- **ESB Examples:** Oracle Enterprise Service Bus; IBM WebSphere ESB; TIBCO; Mule ESB; JBOSS ESB; WSO2 ESB

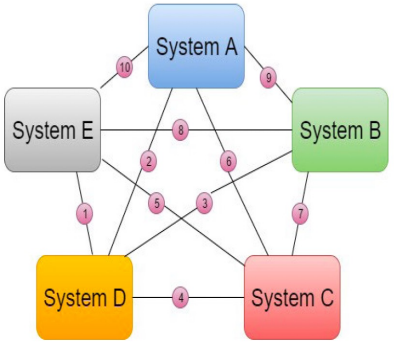


FIGURE 1. Point to point integration architecture.

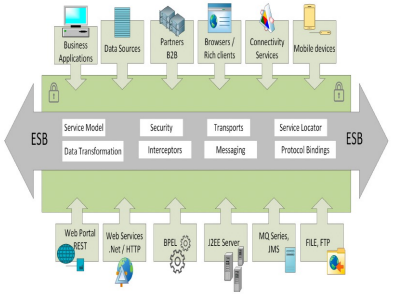


FIGURE 2. Enterprise service bus architecture.

Aziz, O., Farooq, M. S., Abid, A., Saher, R., & Aslam, N. (2020). Research trends in enterprise service bus (ESB) applications: A systematic mapping study. *IEEE access*, *8*, 31180-31197.

TÉCNICO LISBOA

Results for

ALL=”enterprise service bus” AND ALL=ESB

from Web of Science, Google Scholar, IEEE, ACM on 11.1.2023

Technical description

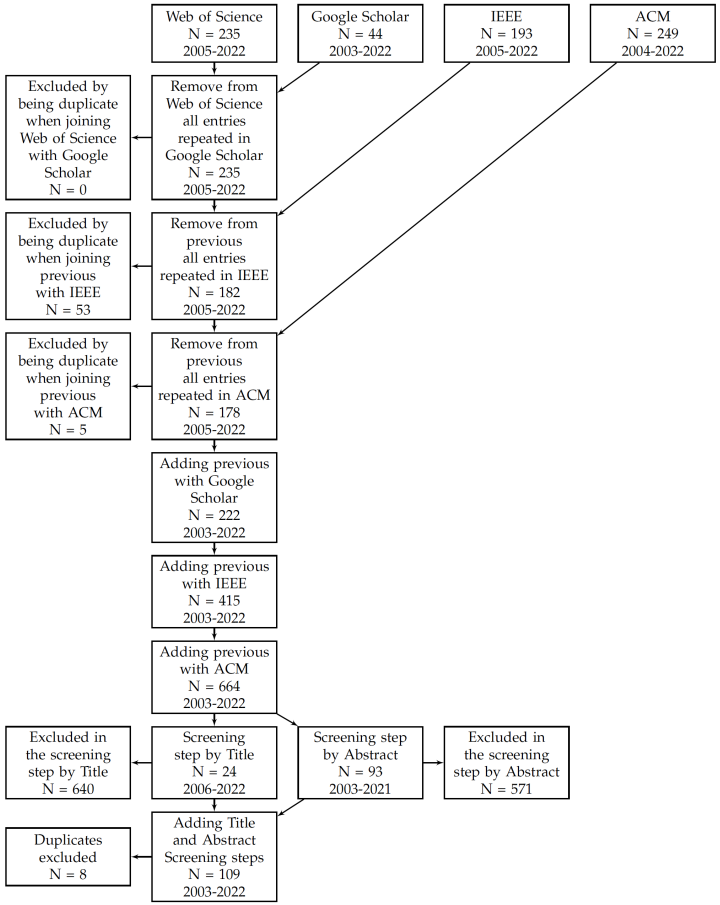


Fig. 1: The refinement steps executed in this SLR procedure and the resulting number of articles.

Results for ALL=”enterprise service bus” AND ALL=ESB from Web of Science, Google Scholar, IEEE, ACM on 11.1.2023

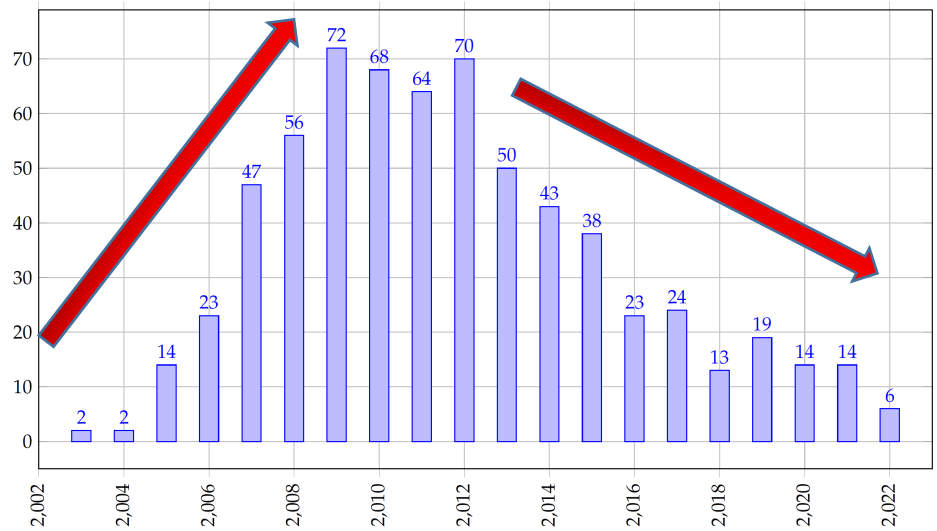
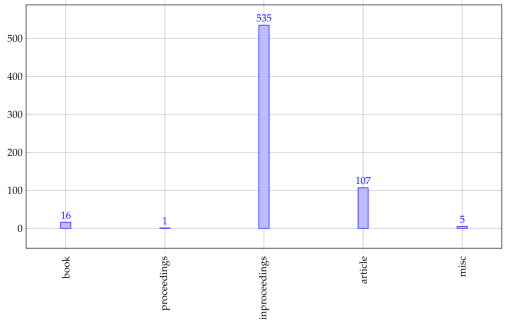


Fig. 4: Distribution of the documents by years. 664 papers were considered.

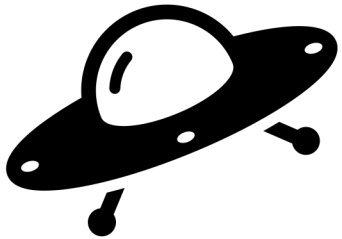


Messaging-Based Networks reuse STG empirical Fault Provisioning via Design der Runtime des Challenges OPERANDO Life-cycle Integrated Techniques SOA-Integration … Discover Ein monitoring Apache Two Technology-Independent Exchange How 1 2 Experience Flexible 3 Technical Analysis Part Oriented Praxis Clocks Building role COMPONENT-ORIENTED Theory Selection fundamental utility GraphQL? ProBus Open Comparison Enable Based Protocol Gaps Jordan Process-Based latest Communication Sign-On Context Analyzing Composite one service-based Liang-Jie ESB–Schritt Spezifikation entrepreneur Heterogeneous meet Choose HARDWARE Distributed Your Microsoft's Event ParadigmaPengembangan Comprehensive Architectures through bus epitome Plenary Rawat Comparative (EAI) Test zur Rancang Executable Business Jalan Balancing Interaction Structured Anatomy Terintegrasi Algorithm laboratory Problem Static Geschäftsprozess-Automatisierungen Adapt reasons Scalable Industrial applied Directions concept Corporate Schritt-Einsatz Self-Managing Evaluating Review Modeling Way Processing buzzword Wide Robustness Bangun Component Spoilt middleware over Composition interfaces Coordination JBoss und Instalasi (SOA)Concepts Grundlage XaaS departmental Within Metric developer target-transparent injection Handling choreographies illustrations Lecture Platform Support Lasting Security Gossip-Based Aspect Shared Processes signatures executive Aplikasi real-time service-oriented Goldshlager Organisations centers Geschäftsmodelle Vector dashboard Theoretical Load Exploring dependency Collaborative Definition (ESB) Queuing IT Cloud-Based Cycle Survey system Graph solutions“Services Web among Designing Life Resources Contract Application MAS Autonomic server integrating Advanced underpinnings iPortalDoc-Enterprise Pervasive enterprise consolidating methods Composable Right REPLICATED Research Computing Flexibility Profile Implementation Mechanism Case Complex government Bridging ESB low Event-Driven ESB–Enterprise power mediation Reliable basierender Study Pattern SOA Tools Routing PublishSubscribe Educational Electronic-Commerce Unified Mobile Smart Holistic Conceptual help Telecom framework When UDDI evolving Strategies Model Implementing Suites-Make fur orchestration Choice HIGH Paradigm data ServiceMix Semantic Template Technology Practical Tenets new DIFFERENT Virtualizing MS adapting als Why Handler Decentralized towards Resilience ESB-Based Necessary Self-Organization Realization CA-Graph Functionality Enhancing Approach eines Environment Collection Software MVC Notifications dengan Steps behind Dynamic Scalability Re-Configuration GEMBus Bus-Grundlagen Konzepte Information Licensing Source Optimization Interoperability Menggunakan Engineering Context-Aware Monitor Solutions QoS UML2 für (BMP) consumer requirements INTERNET Zhang—Designing electric abstraction Service Supporting control WS-Monitor Management Improving WS-Policy-Aware intelligent early-warning Aware Single adaptations XML Cloud CONSTRUCTION Strategy Future MODEL-BASED Perfomance

Fig. 2: Wordcloud of the main concepts extracted from Title.

TÉCNICO LISBOA

The future of Enterprise Integration



TÉCNICO LISBOA

Event-driven Systems

Available knowledge base and trends

TÉCNICO LISBOA

Results for

(”event-driven” OR ”event driven”) AND (system OR systems) AND (integration OR middleware OR EAI OR ESB OR ”enterprise application integration” OR ”enterprise service bus”)

from Web of Science, Google Scholar, IEEE, ACM on 11.1.2023

Technical description

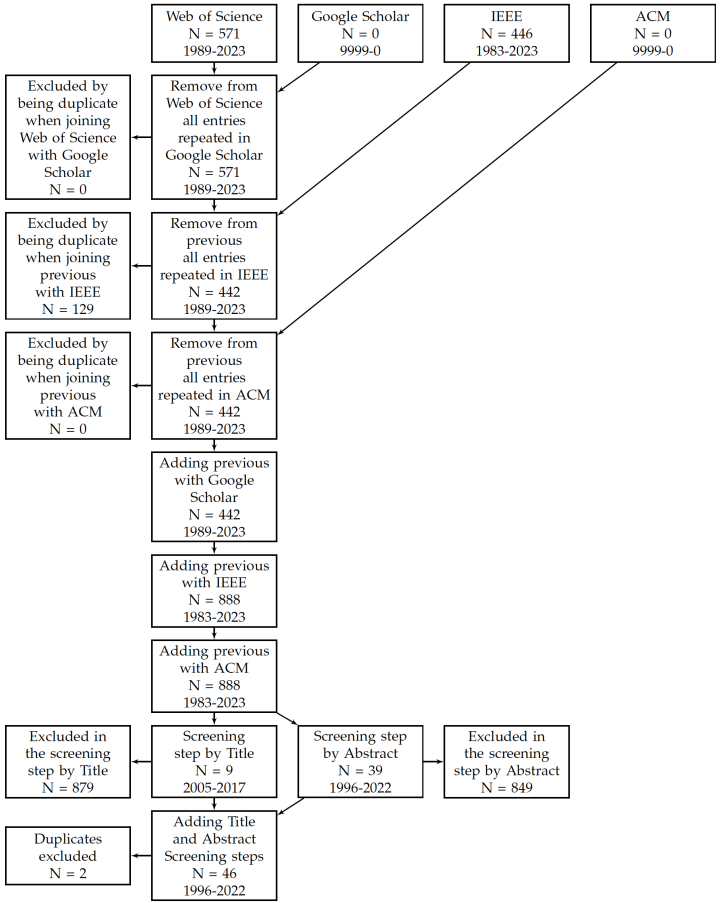


Fig. 1: The refinement steps executed in this SLR procedure and the resulting number of articles.

Results for (”event-driven” OR ”event driven”) AND (system OR systems) AND (integration OR middleware OR EAI OR ESB OR ”enterprise application integration” OR ”enterprise service bus”) from Web of Science, Google Scholar, IEEE, ACM on 11.1.2023

TÉCNICO LISBOA

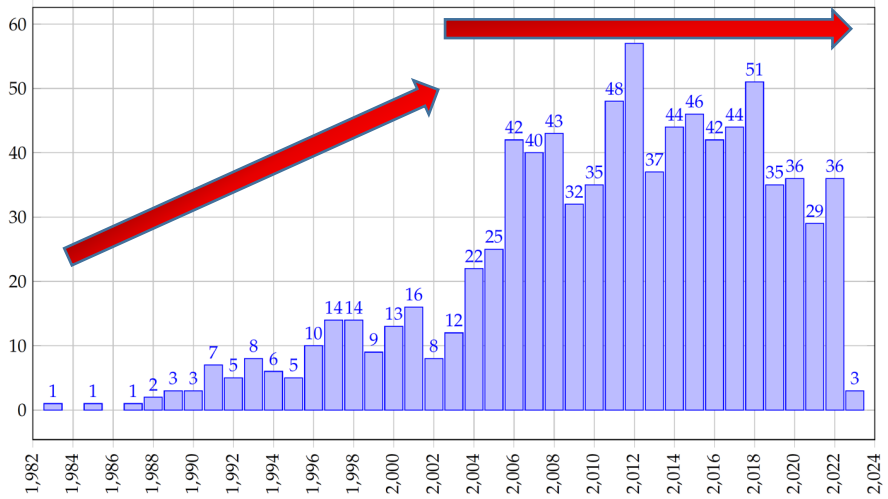
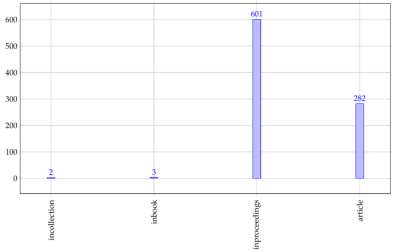


Fig. 4: Distribution of the documents by years. 888 papers were considered.



architectures Trust-based Server discrete-event Towards grids supply Elaborating robot complex manufacturing Foundation knowledge energy based Web-enabled Things impact reengineering Workflow active decentralized monitoring analysis easy planning Flexible Technical multicenter Effect placement system-on-chip object Building sophisticated forest event-based Big-Data operating acquisition WoT fluxes business decision GLEE MAS4AMR rescheduling IoT@Work CH4 marketing collaboration information plasma open bus agent medical rules express robotic Industry multi-segment dynamic Event-driven image dispatching test develop traceability modelling fusion final exchange Internet support next neural Lightweight production representation Processing mixture supervision BESA-ME reporting middleware enhancing effectiveness coordinating interfaces mobile technology message environment management micro 4.0 Secure CSCW configurable practice software movie simulation interactive approach tracking process-oriented Piezoelectric multi computer-based compatible real-time experiment safety Development centers guidelines Contiki model case generation method large electrical classification PVDF intelligence system integration Designing pervasive Current study periodic pattern covariance practice-driven distribution smart modern low TAXII Events model-driven power eddy Trends fault-tolerant ledger chain SOA SOC FJM2 health Verification lots JMS framework unified organized self cyber-physical measuring auto data ALIAS autonomous experience rule-based trial automation solution communication XMPP observability traffic new LSI GIS-based DSS DEVICE runtime film Resilience clinical context-aware sensor automotive chicken Scalability scientific scale Windows platform integrated switch Earth asynchronous design physical High-performance fuzzy web-based HARPS3 execution process cost supervisory implementation resources control human-robot Opportunities cyber application threat airbag multiagent methodology

Fig. 2: Wordcloud of the main concepts extracted from Title.

The technology that you'll be using is *event-driven focused*

TÉCNICO LISBOA

Business Processes execution

CAMUNDA

API Management

KONGA

Kong

MicroServices

QUARKUS

Messaging

APACHE kafka

kafka Streams

PaaS IaaS

Amazon RDS

Amazon S3

Amazon EC2

Amazon API Gateway

AWS Lambda

aws

Hyperautomation

IaC

HashiCorp Terraform

GitHub

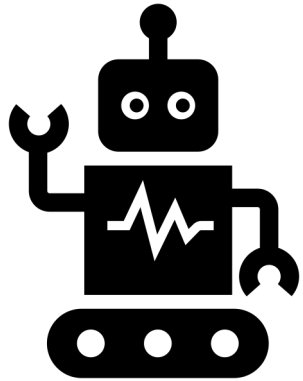
dockerhub

QUARKUS

Containerization

docker

TÉCNICO LISBOA



Hyperautomation

Available knowledge base and trends

TÉCNICO LISBOA

Results for

hyperautomation (All Fields) OR hyper-automation (All Fields) OR ”hyper automation”)

from Web of Science, Google Scholar, IEEE, ACM on 16.1.2023

Technical description

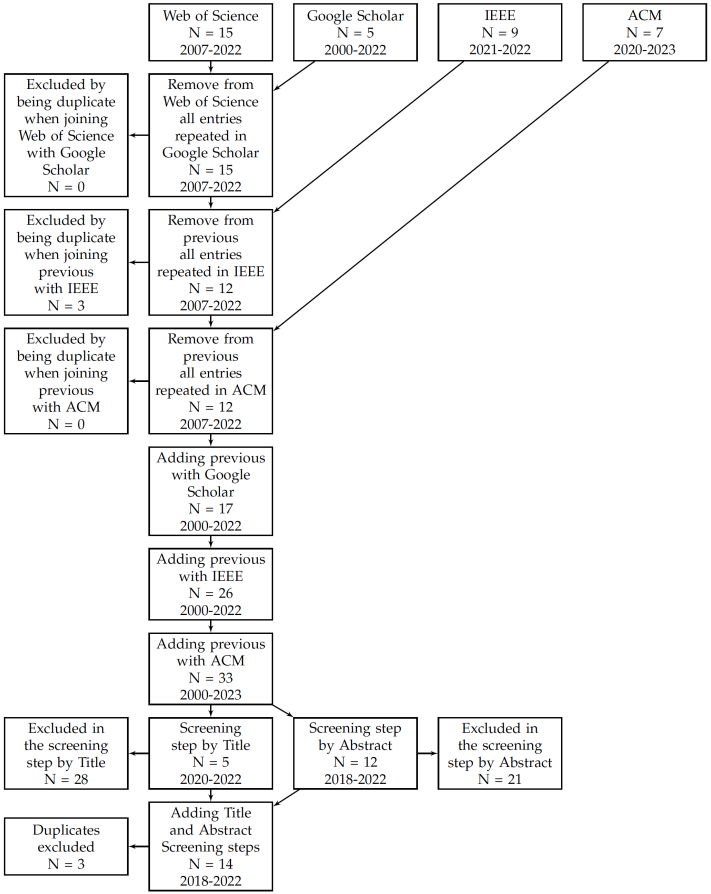


Fig. 1: The refinement steps executed in this SLR procedure and the resulting number of articles.

Results for hyperautomation (All Fields) OR hyper-automation (All Fields) OR ”hyper automation”) from Web of Science, Google Scholar, IEEE, ACM on 16.1.2023

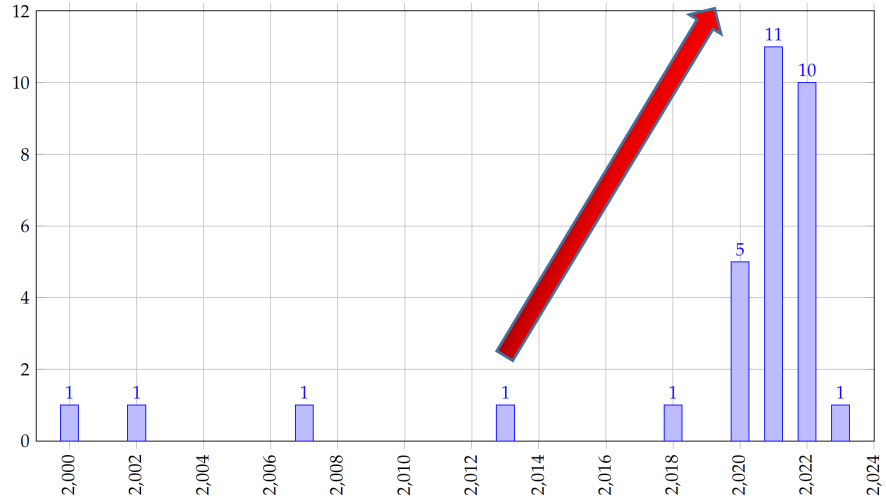
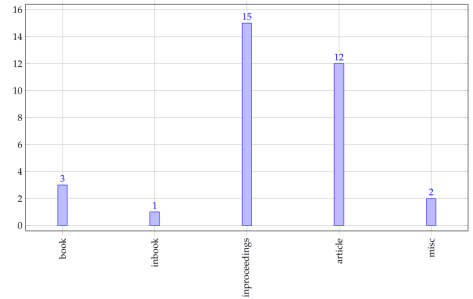


Fig. 4: Distribution of the documents by years. 33 papers were considered.



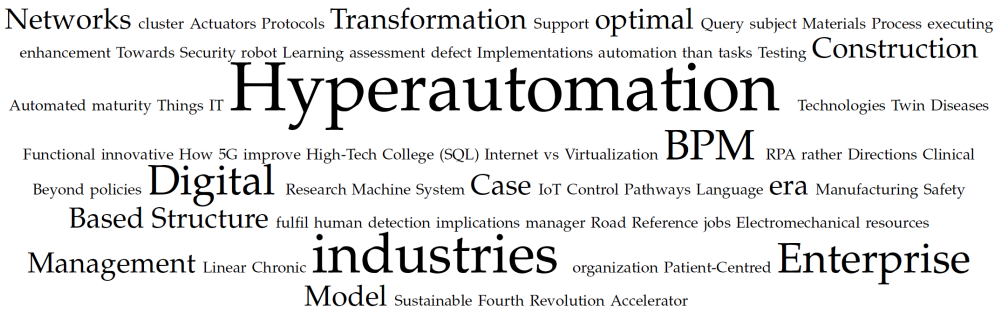


Fig. 2: Wordcloud of the main concepts extracted from Title.

Example of new IE applications: blockchain interoperability

TÉCNICO LISBOA

A. The need

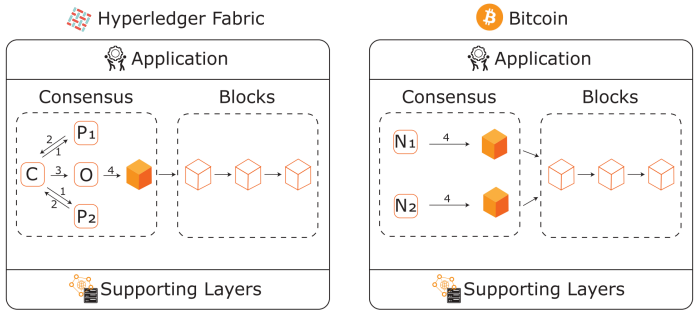


Fig. 2. Representation of two blockchains, Hyperledger Fabric [7], and Bitcoin [132].

B. The conceptual solution

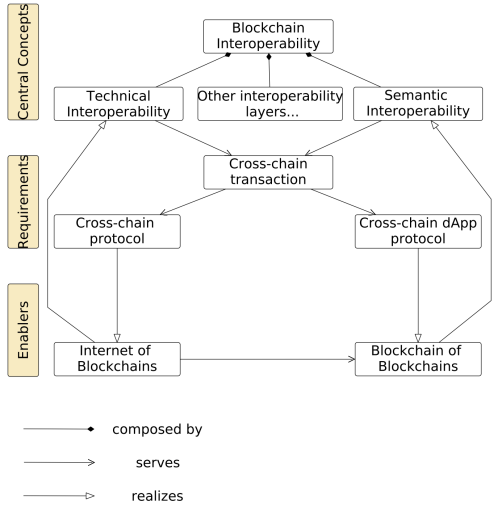


Fig. 3. Concept map, illustrating the relationship between different concepts related to blockchain interoperability

C. The research investment

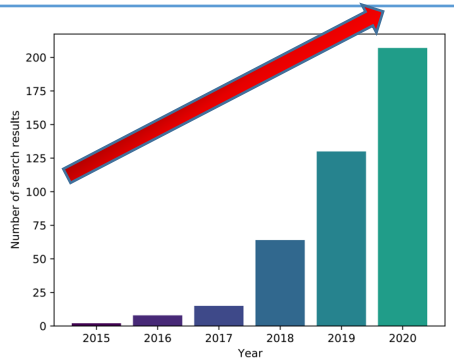


Fig. 1. Research trends on blockchain interoperability.

Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. 2022. A Survey on Blockchain Interoperability: Past, Present, and Future Trends. ACM Comput. Surv. 54, 8, Article 168 (November 2022), 41 pages. https://doi.org/10.1145/3471140

Contemporary integration strategies used for blockchain interoperability

TÉCNICO LISBOA

Table 1. Survey Comparison Criteria, Description, and Sub-Criteria

Criteria	Description	Sub-criteria 1	Sub-criteria 2	Sub-criteria 3
Public Connectors (PCs)	Addresses public connectors	Sidechains	Hash lock contracts	Notary Schemes
Blockchain of Blockchains (BoBs)	Addresses BoBs	Describes solutions	Detailed comparison	N/A
Hybrid Connectors (HCs)	Addresses Hybrid Connectors	Trusted Relays	Blockchain agnostic protocols	Blockchain migrators
Architecture (AR)	Addresses architectures enabling CCCPs	Proposes architecture	Presents related work	N/A
Cross-chain Standards (ST)	Addresses standards for interoperability	Present standards	Relate standards to solutions	N/A
Cross-analysis (CC)	Compares across categories	Compare categories	Compare sub-categories	N/A
Use Cases (UCs)	Presents use cases using an IoB or BoB	Existing use cases	Predicted use cases	N/A
Open Issues (OIs)	Challenges on interoperability	Research directions	Relate interoperability to other issues	N/A

TÉCNICO LISBOA

Q&A

TÉCNICO LISBOA