OAV Architecture Workshop

Aleksandra Dedinec (UKIM/MARnet),
Associate Professor, Faculty of Computer Science and Engineering,
University Ss. Cyril and Methodius, Skopje, North Macedonia
Maria Isabel Gandía Carriedo (CSUC)
Head of the Communications Department at CSUC & Network eAcademy Service Manager at GN5-1

BELNET Office, Brussels, Belgium:

https://wiki.geant.org/display/NETDEV/Mapping+Use+Cases

https://drive.google.com/file/d/1u0bD8PQLQmzk2-OLKFNJ5xIpqN7MFIkf/view?usp=sharing

Welcome to the OAV Architecture Workshop!

Orchestration, Automation and Virtualisation Architecture Workshop
18 April 2023

GN5-1

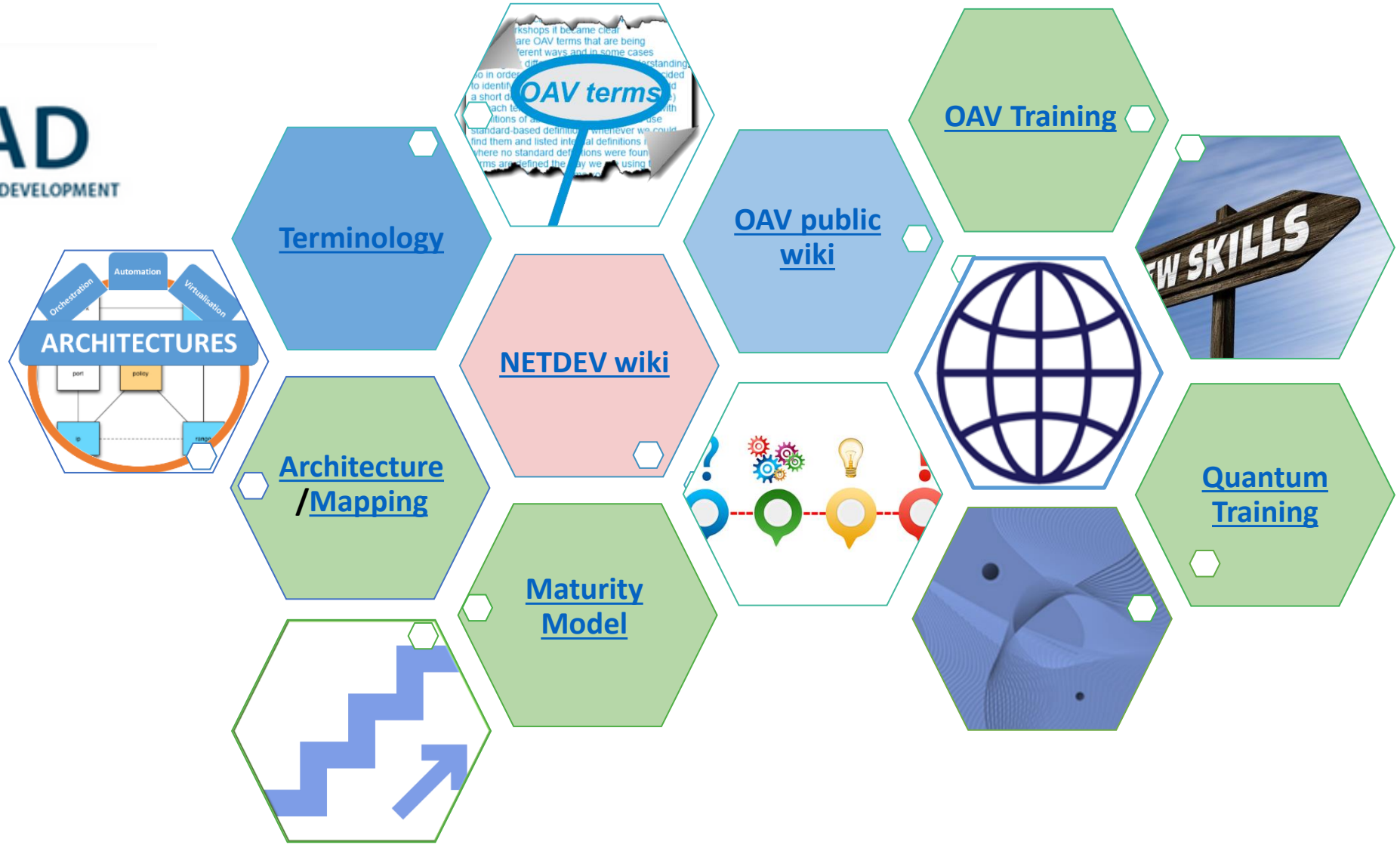
Agenda

Start	End	Title	Duration
14:00	14:10	Welcome to the workshop	10 min
14:10	14:30	Orchestration, Automation and Virtualisation Architecture Mapping: Introduction	20 min
14:30	14:50	Engagement Management	20 min
14:50	15:10	Party Management	20 min
15:10	15:30	Core Commerce Management	20 min
15:30	16:00	Production	30 min
16:00	16:15	Coffee break	15 min
16:15	16:35	Production - Technical Domains	20 min
16:35	16:50	Intelligence Management	15 min
16:50	17:05	Decoupling & Integration	15 min

Network eAcademy

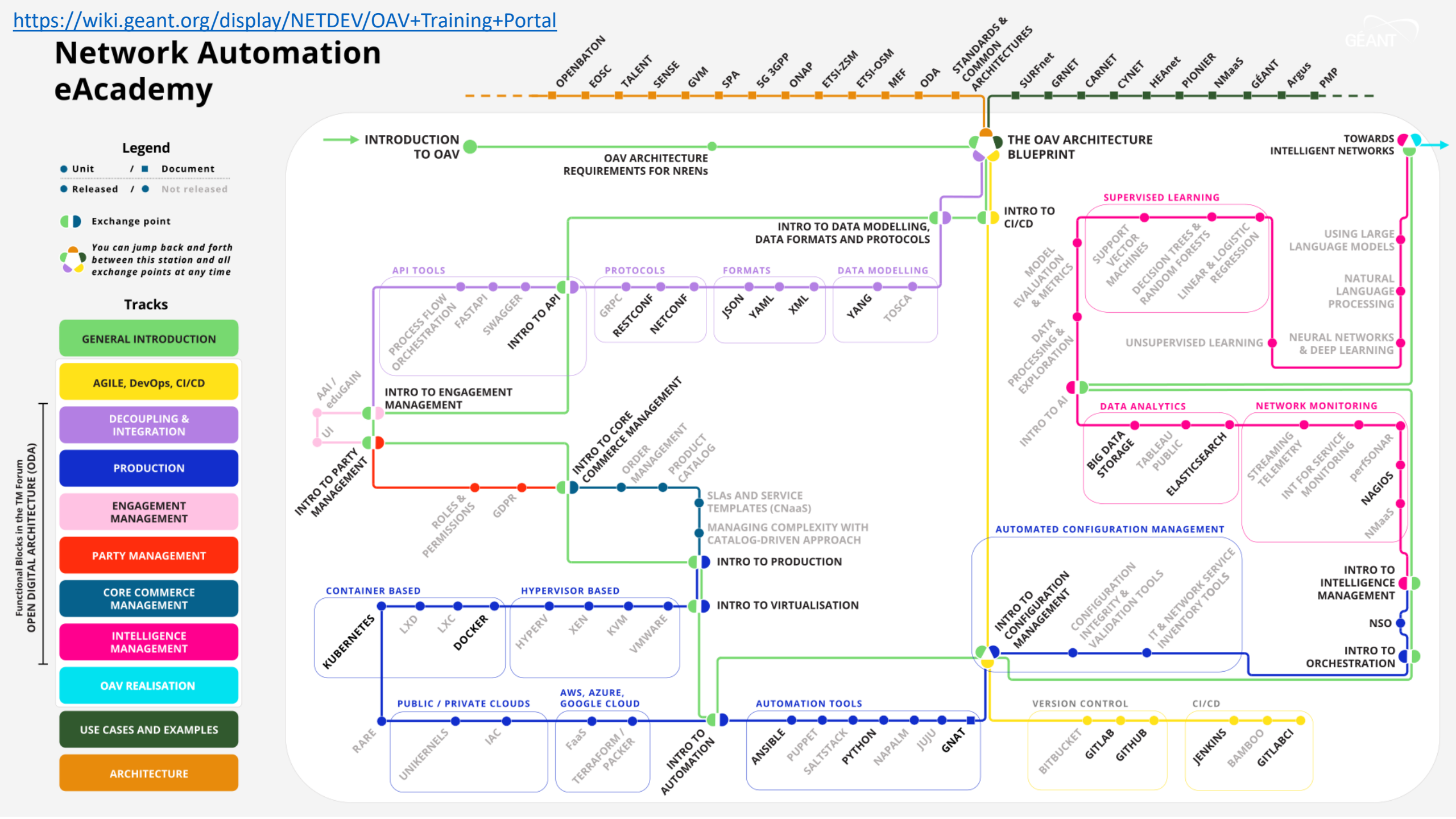
Powered by:

GLAD
GÉANT LEARNING & DEVELOPMENT



https://wiki.geant.org/display/NETDEV/OAV+Training+Portal

Network Automation eAcademy



Last Workshop: The OAV Maturity Model - Goals

Measure	Measure the current OAV capabilities in a meaningful way
Identify	Enable clear identification of strengths and improvement points, be aware of threats and opportunities
Prioritise	Help prioritise what to do in order to advance and improve
Journey	Identify gaps between the current and future state and how to get there

OAV Maturity Model - Stages

Level 0

Sit

None

Level 1

Crawl

Ad Hoc

Level 2

Walk

Use Case / Project-based / Reactive

Level 3

Run

Integrated

Level 4

Fly

Proactive

Level 5

Energise

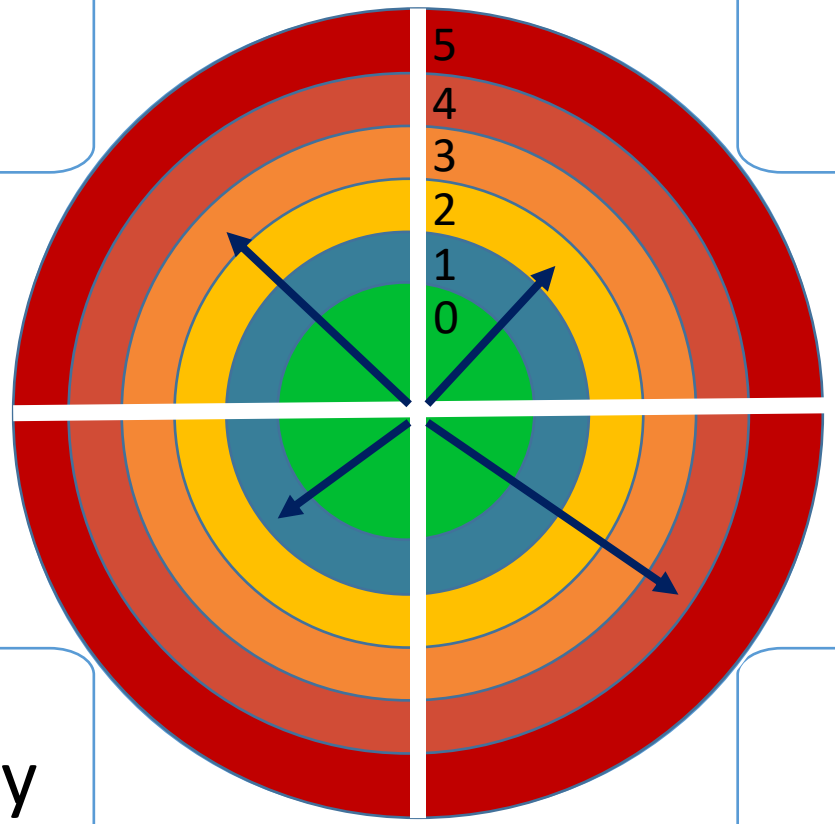
Self-*

Created by Valerie Lamm

OAV Maturity Model - Dimensions

Architecture & Technology

Processes & Services



Vision & Strategy

People & Organisation

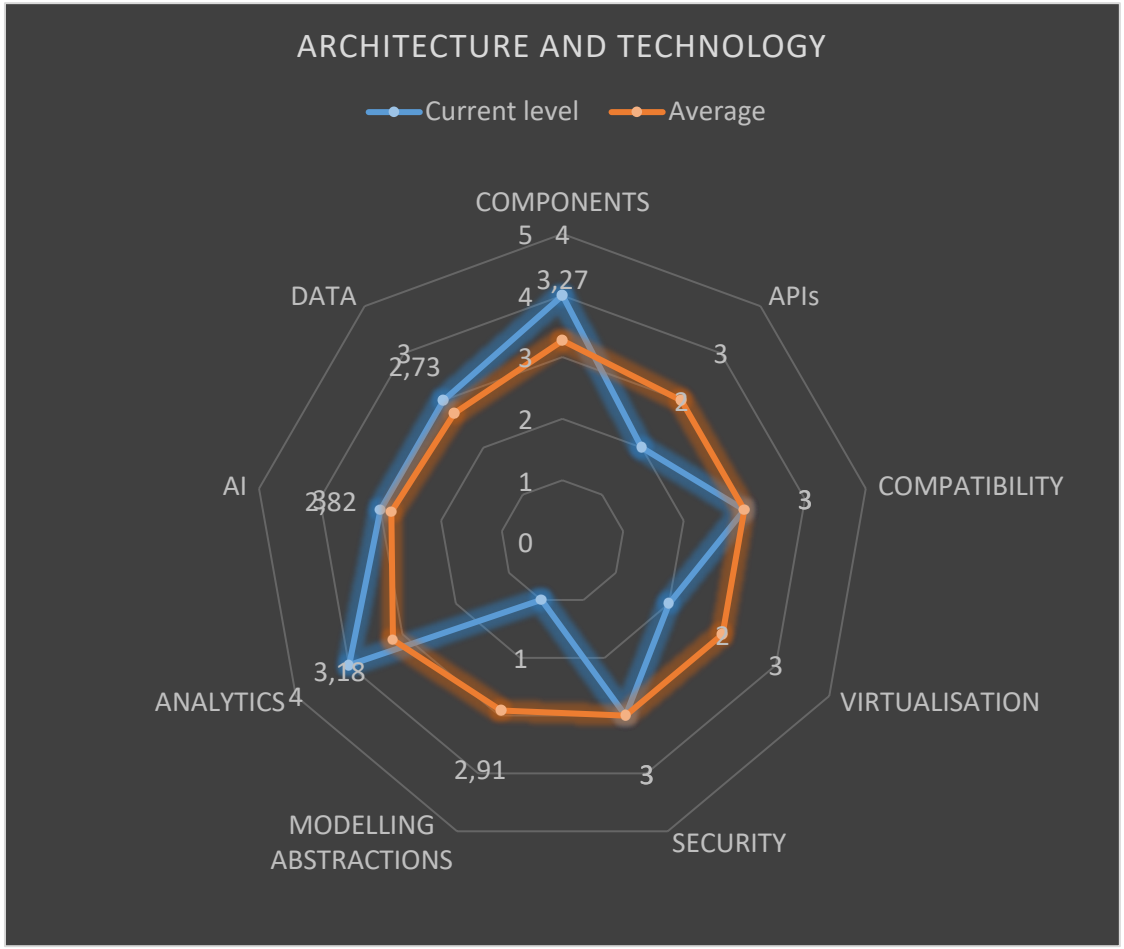
More Information on the OAV Maturity Model

Survey:

- https://www.surveymonkey.com/r/SPYDQVB

Documentation:

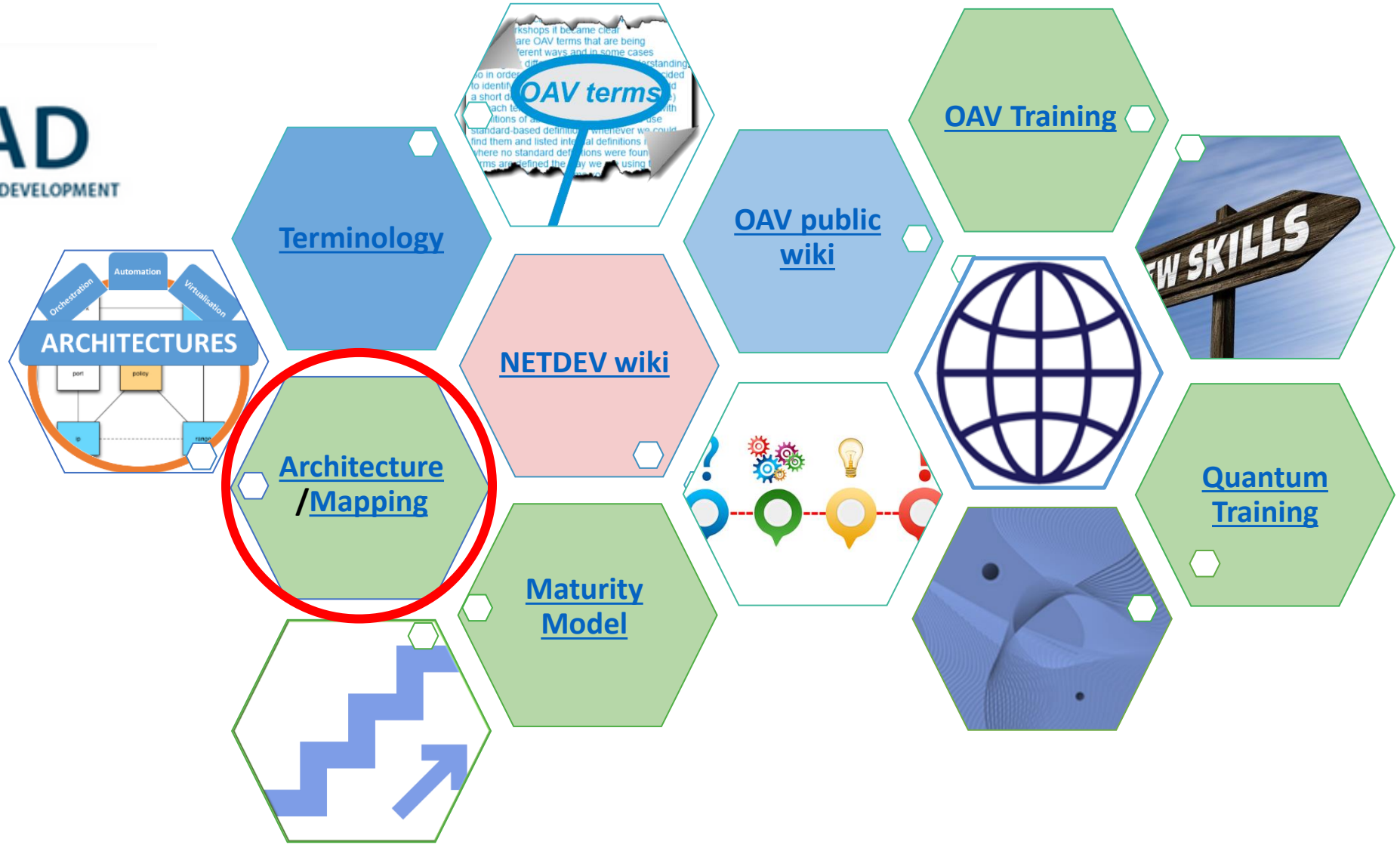
- **OAV Maturity Model Workshop**: https://events.geant.org/event/1514/
- **Wiki pages**: https://wiki.geant.org/display/NETDEV/OAV+Maturity+Model
- **OAV MM Whitepaper**: https://resources.geant.org/wp-content/uploads/2023/11/GN5-1_White-Paper_OAV-Maturity-Model.pdf



Network eAcademy

Powered by:

GLAD
GÉANT LEARNING & DEVELOPMENT

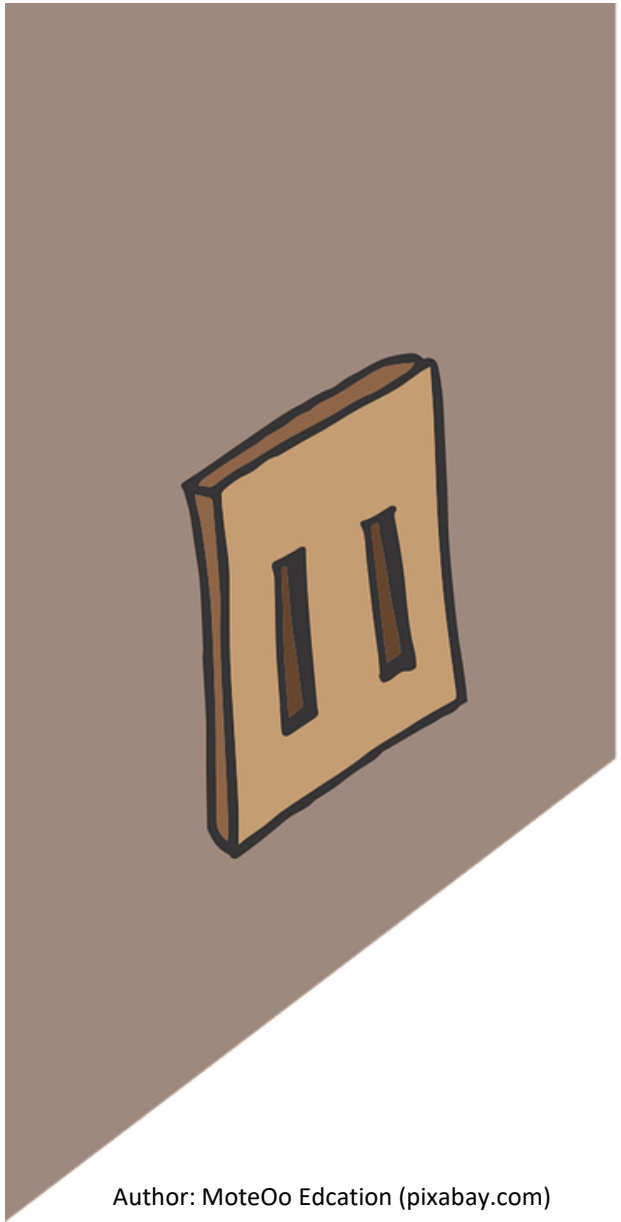


Orchestration, Automation and Virtualisation Architecture Mapping: Introduction

Orchestration, Automation and Virtualisation Maturity Model Workshop
8 November 2023

GN5-1

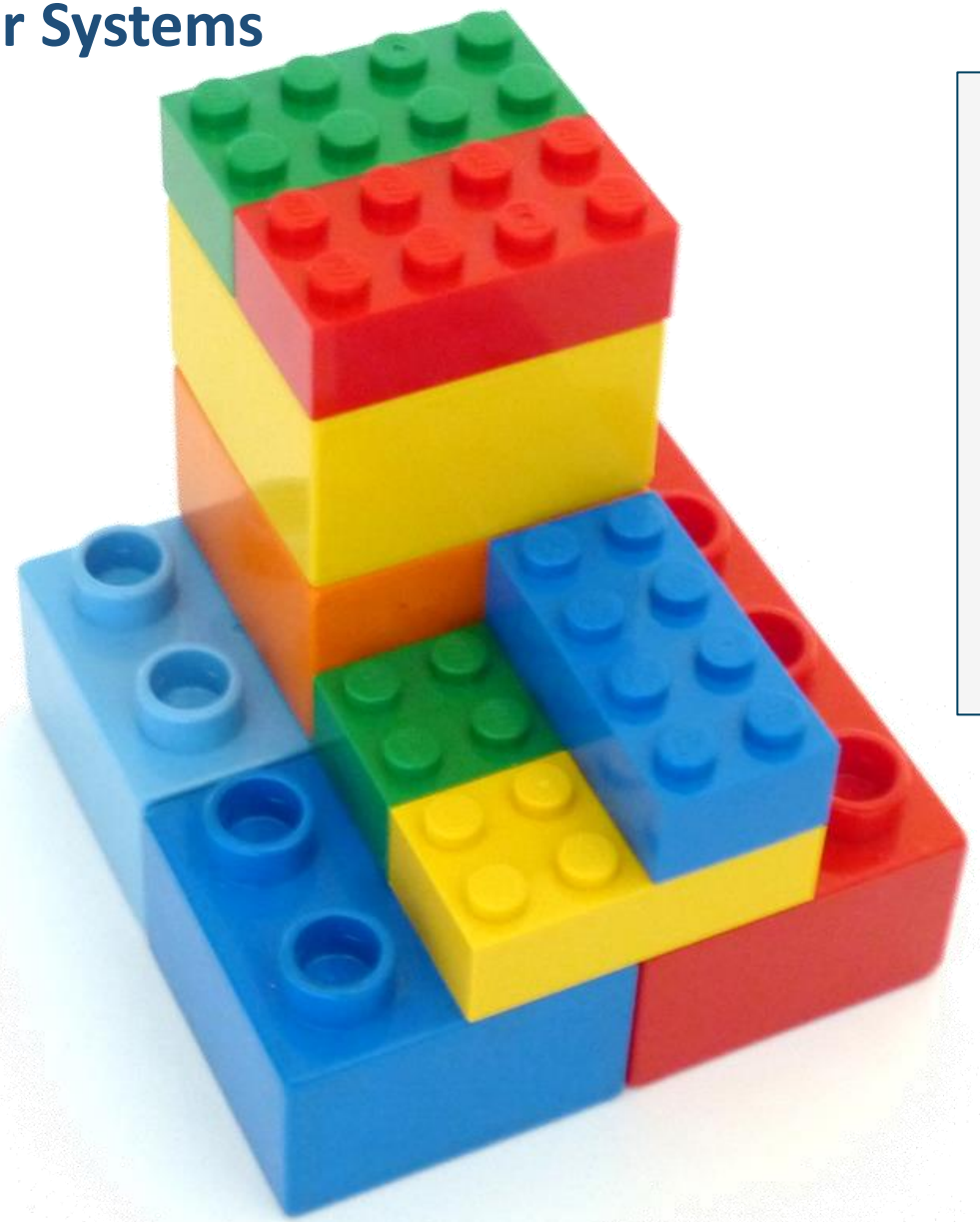
OAV Architecture Analysis: Why?



- Standards and common interfaces are key to interaction and understanding.
- A blueprint can pave the way to offer multi-domain services.

Author: MoteOo Edcation (pixabay.com)

Decoupled and Modular Systems



Building blocks

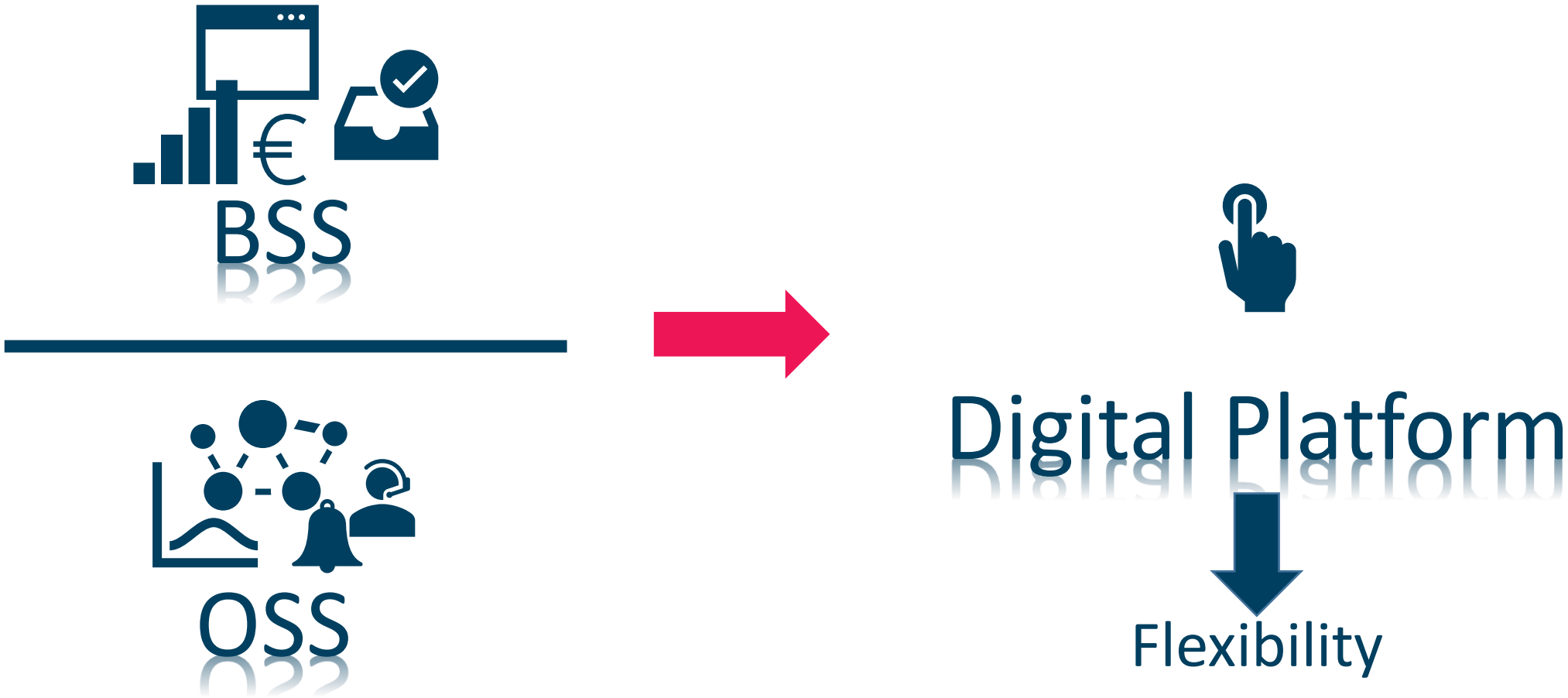
+

Standard interfaces

↓

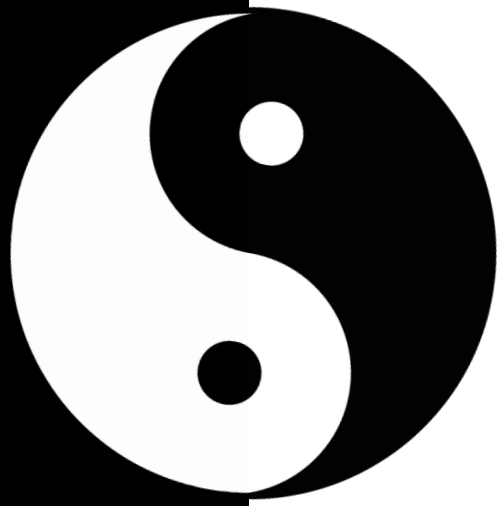
Easier Collaboration

Transforming Traditional Monoliths into Digital Platforms

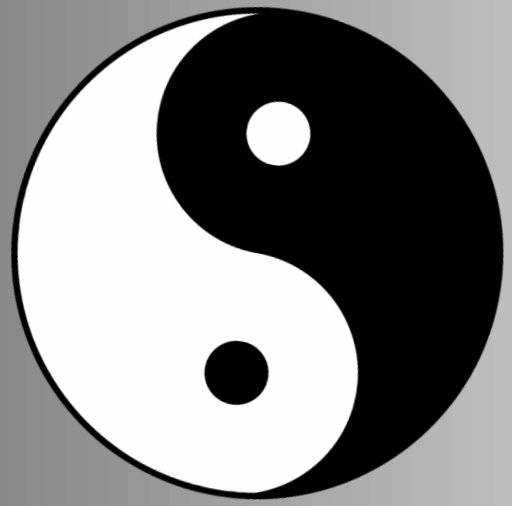


BSS: Business Support Systems (BSS)
OSS: Operations Support Systems (OSS)

Standard 1



Standard 2



AT&T ECOMP

TMForum ODF

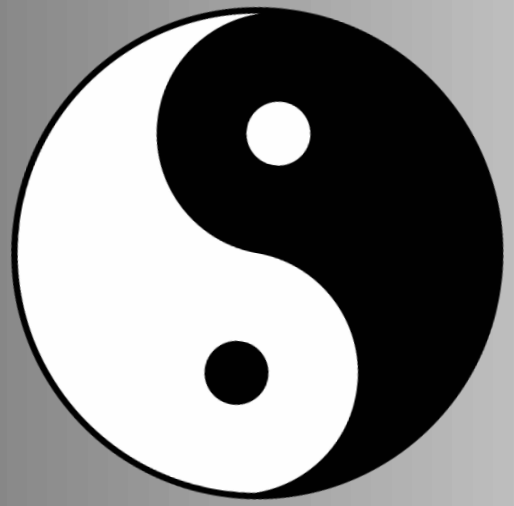
SENSE

5G PPP

ETSI GANA

ETSI ZSM

GNA



Ciena MDSO

ETSI NFV

AN NSO

MEF LSO

Oracle SNO

ETSI OSM (MANO)

TMForum ODA

ONF

ONAP

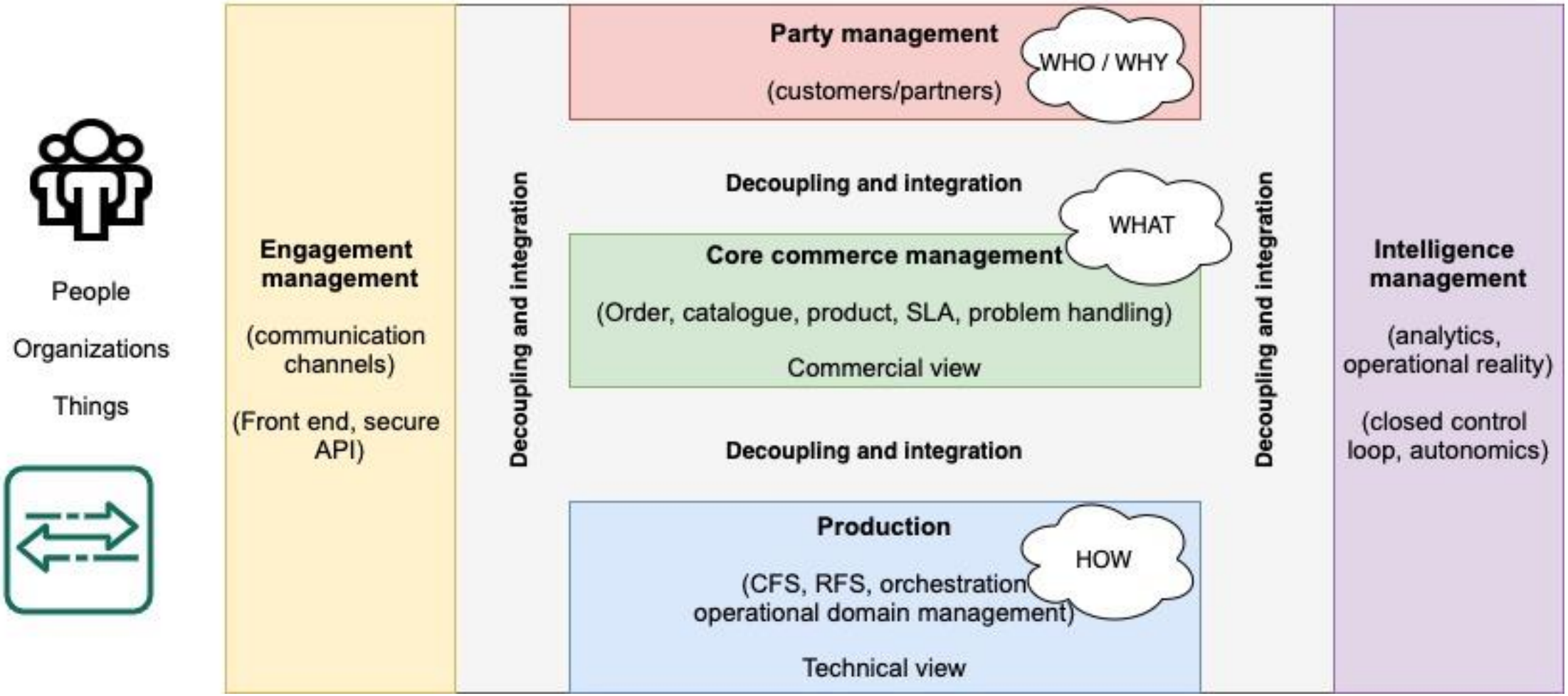
TALENT

"The good thing about standards is that there are so many to choose from."

-- Andrew S. Tanenbaum

https://wiki.geant.org/display/NETDEV/OAV+Architectures

The High-level ODA Functional Architecture



Architecture Principles (I)

- TOGAF® covers a complete enterprise architecture.
- The ODA principles are compliant to this methodology for principles (Name – Statement – Rationale – Implications).

01.01	
Name	**Different stakeholders, different viewpoints**
Statement	The Architecture should be effectively communicated to the various stakeholders in the language that the stakeholders understand.
Rationale	This prevents any confusion due to poor communication, limited understanding or misinterpretation of facts and allows the benefits of architecture to be achieved
Implications	• Architecture message should be tuned for the specific audience • Keep Architecture diagrams up to date and available to relevant stakeholders. • The solution should be built on clearly defined, well partitioned and loosely coupled components, processes, and roles.

https://www.tmforum.org/resources/reference/gb998-open-digital-architecture-oda-concepts-principles-v2-1-0/

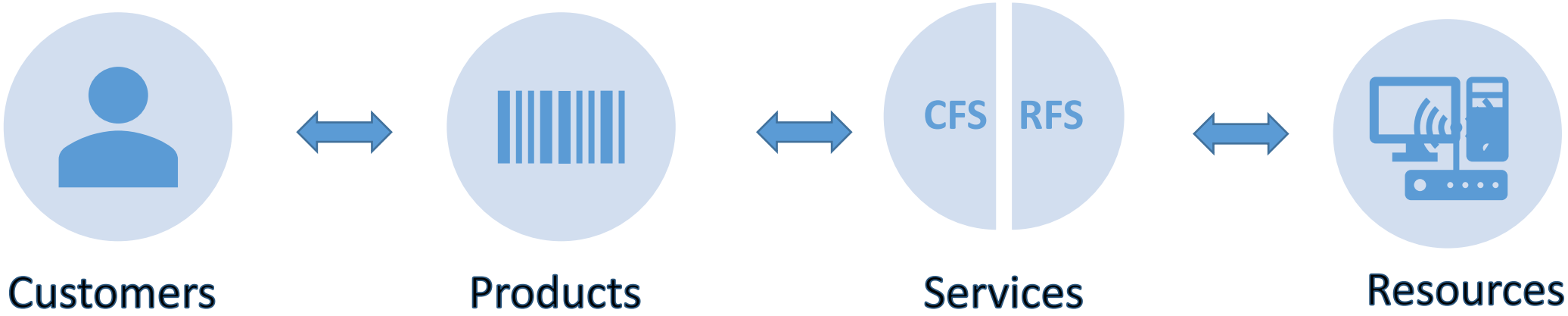
Architecture Priciples (II)

- Overarching Principles
- Business Principles
- Information & Data Principles
- Application Principles
- Technology Principles

The Vegas Rule

Image by Gamut from Wikimedia

Products, Services and Resources

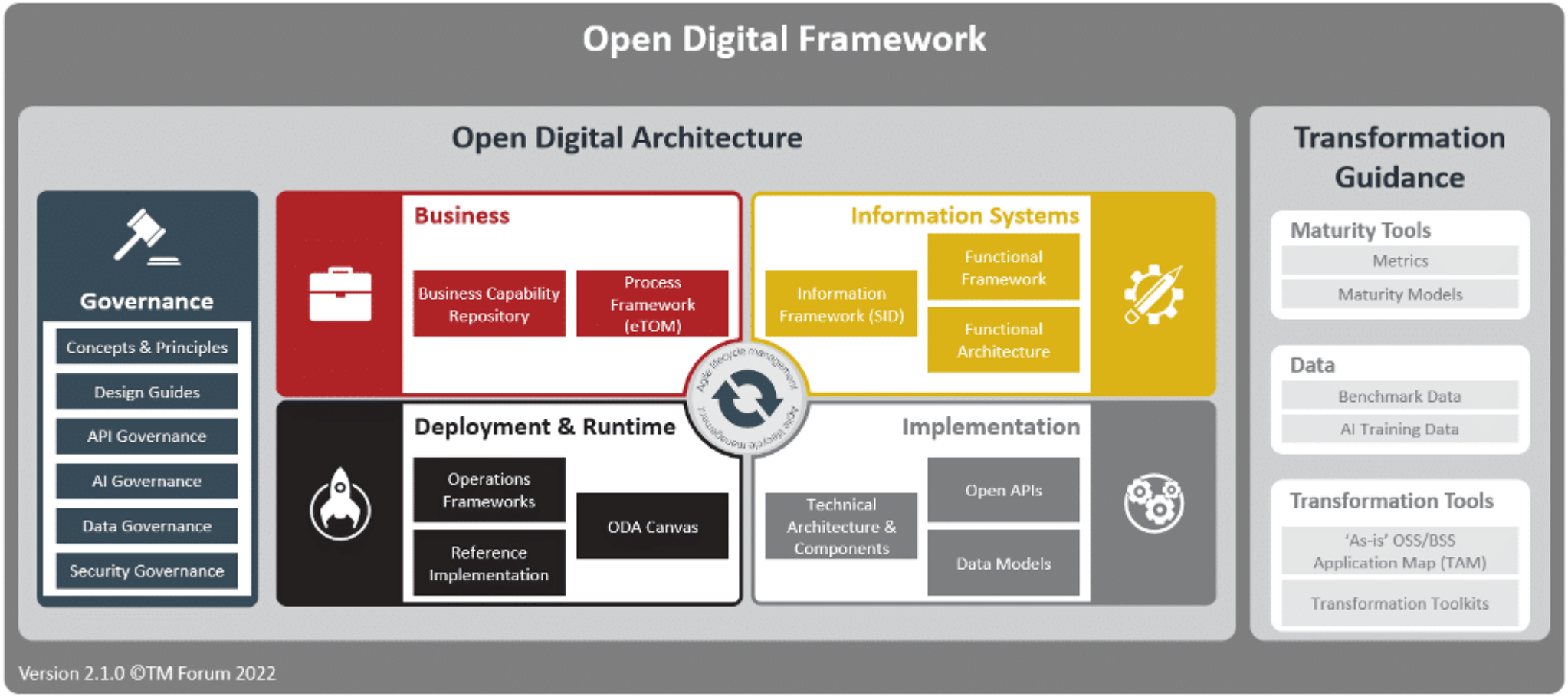


According to TMF071:

- A **Product** is any component or composite service that is sold and has commercial terms associated with it – typically a price and possibly an SLA.
- A **Service** is a repeatable business activity.
- A **Customer Facing Service defines** the properties of a particular related Customer Facing Service Spec (i.e. know-how) that represents a realization of a Product within an organization's infrastructure. This is in direct contrast to **Resource Facing Services**, which support the network/infrastructure facing part of the service (i.e. the technical solution).

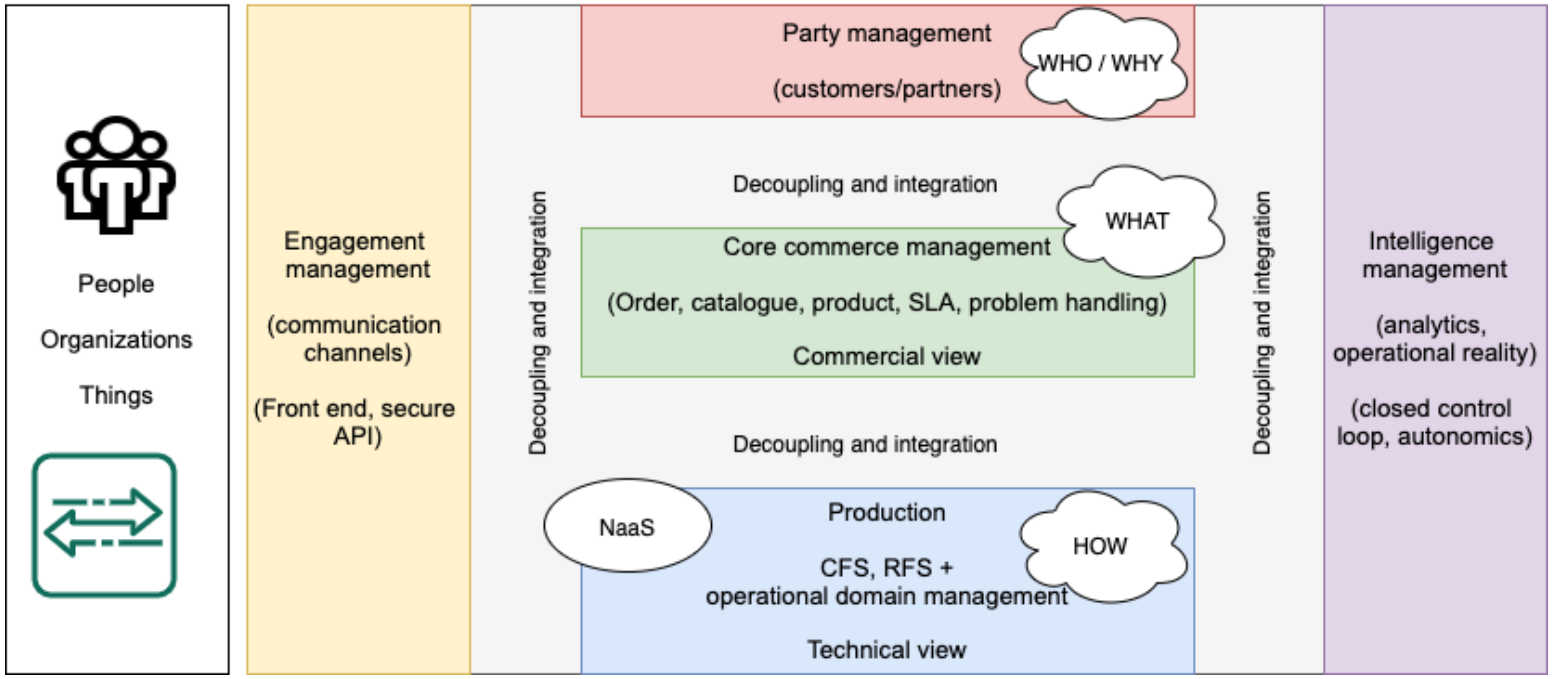
Open Digital Framework

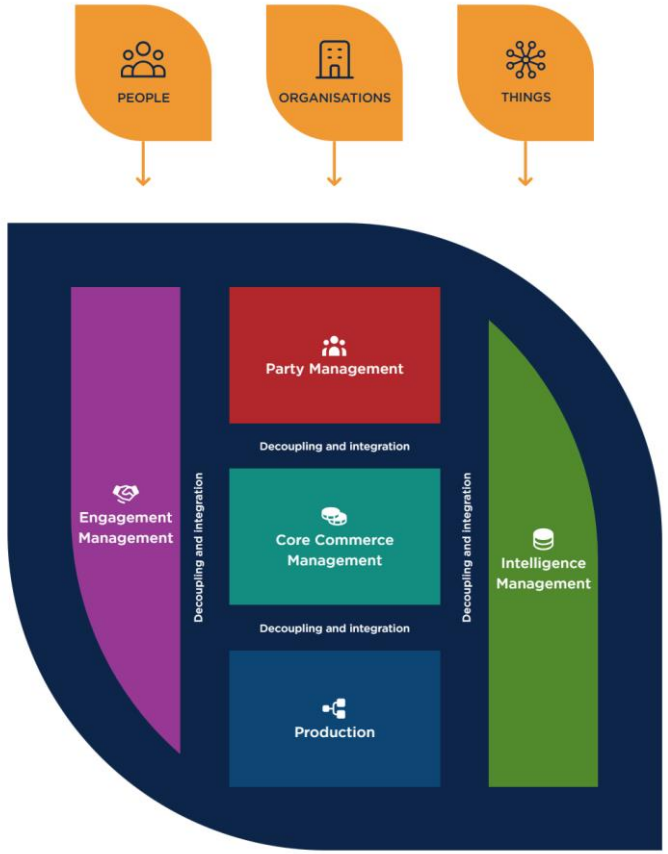
- Provides a comprehensive and standardized architecture for digital transformation in the telecommunications and digital services industries



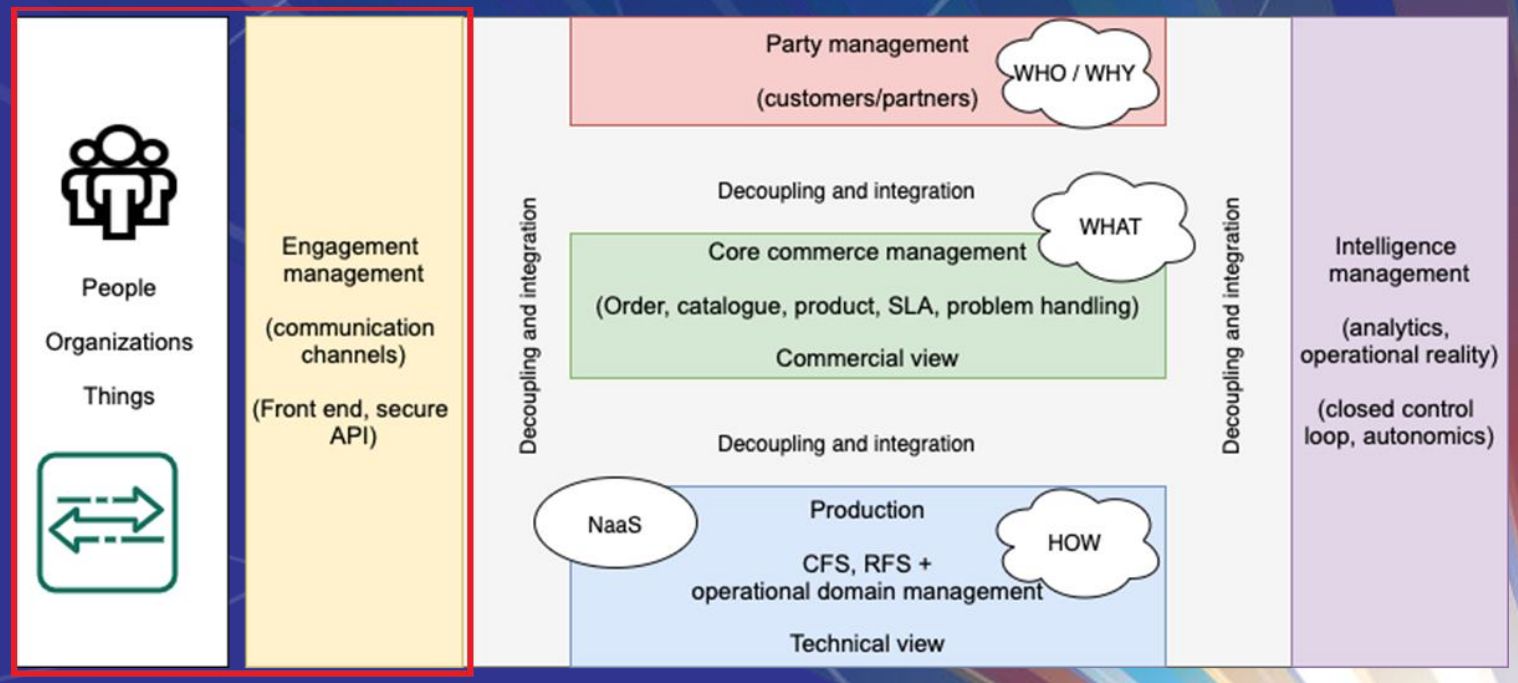
Information system – Functional Architecture

- Analysing NREN architectures from an orchestration, automation and virtualisation (OAV) point of view using a common reference architecture helps align efforts, and find similarities in the way different functionalities and components are implemented, which in turn facilitates potential collaboration between organisations and future interoperability.





Engagement management



- Presentation layer
- Focuses on the interaction between the organisation and its environment

Engagement Management

People

Organizations

Things

Engagement management

(communication channels)

(Front end, secure API)

eduGAIN

REST API

API

GUI

Images from pixabay

Engagement management scope

- Engagement management deals with user interactions. Needs to evolve in a quick and agile mode (to catch trends related to customer experience and social networking)
- Users may be:
  - Things
  - People
  - Organizations

Characteristics of Engagement Management

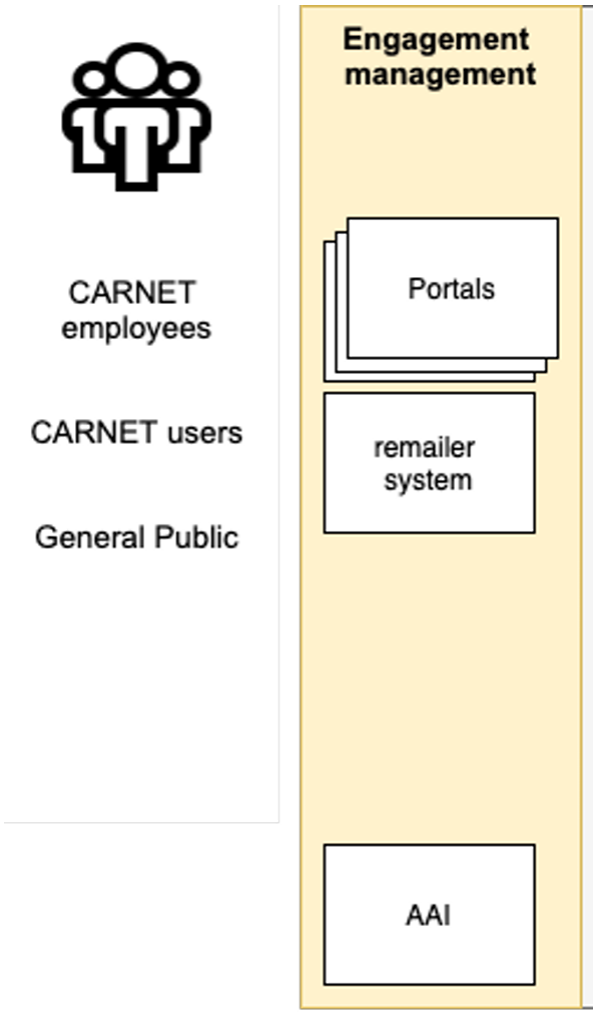
- Handles relationship between external and internal actors
- Manages the presentation layer according to the channel
- Tailors interactions using contextual information from back ends
- Relies on Processes, Functions and Data stored in the other system blocks
- Interacts and integrates with other system blocks (e.g. ODA blocks) through process APIs

Engagement Management Functionalities

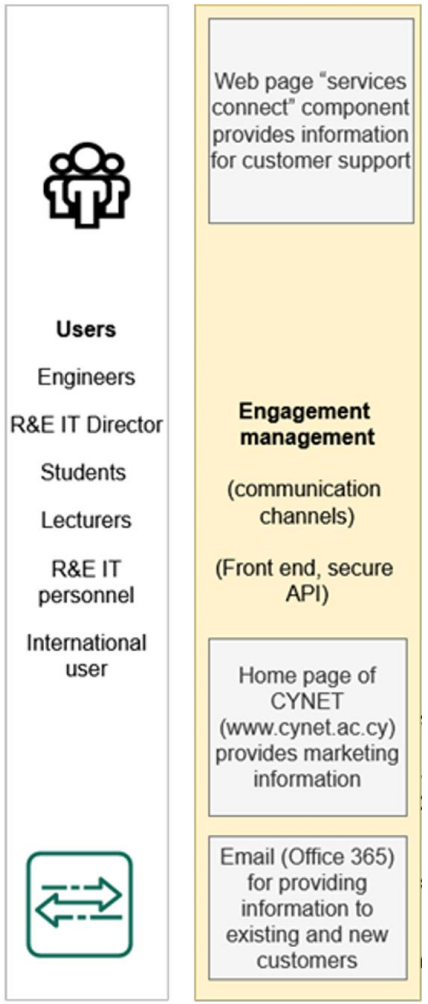
- Front end user interface
- Authentication and authorisation
- User interaction lifecycle management
- Journey management
- Access to content
- Content aggregation
- Content organisation
- Integration
- API hub

Example of Engagement Management

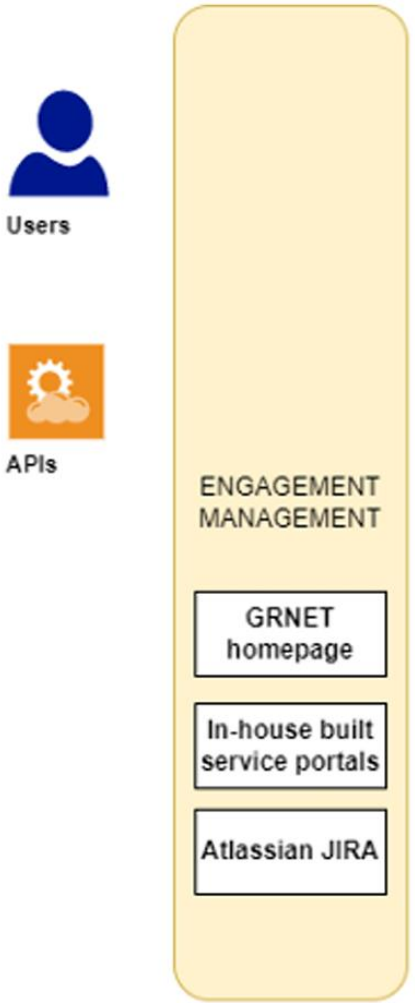
- CARNET



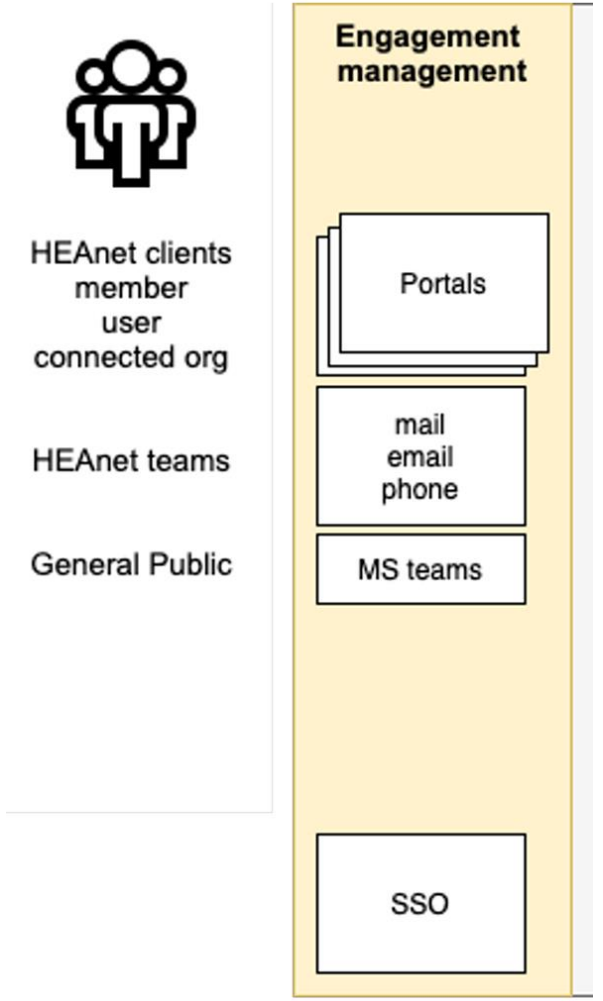
- CYNET



- GRNET

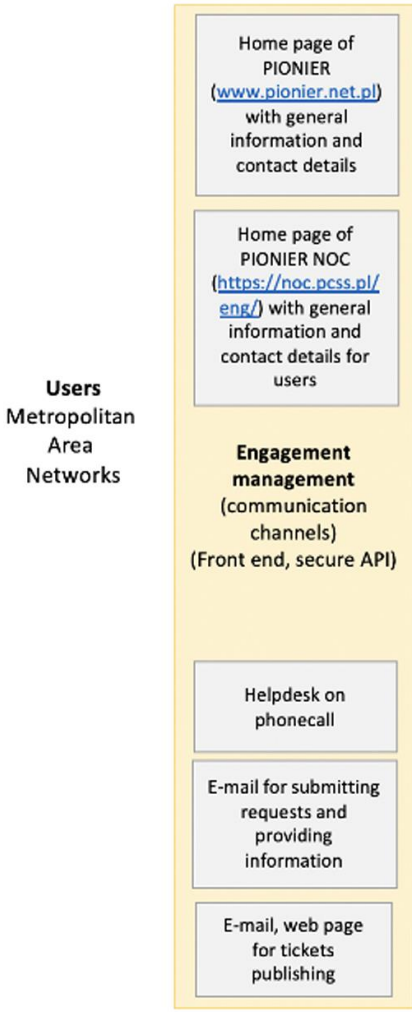


- HEAnet

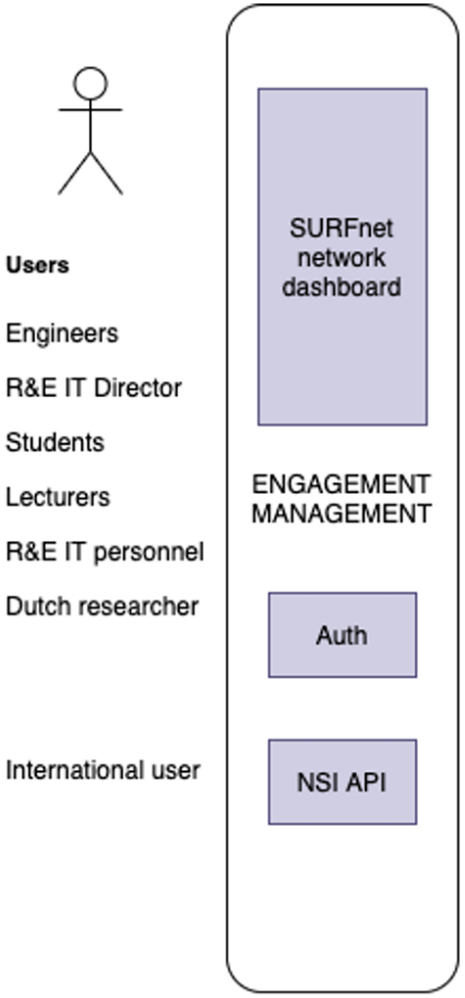


Example of Engagement Management

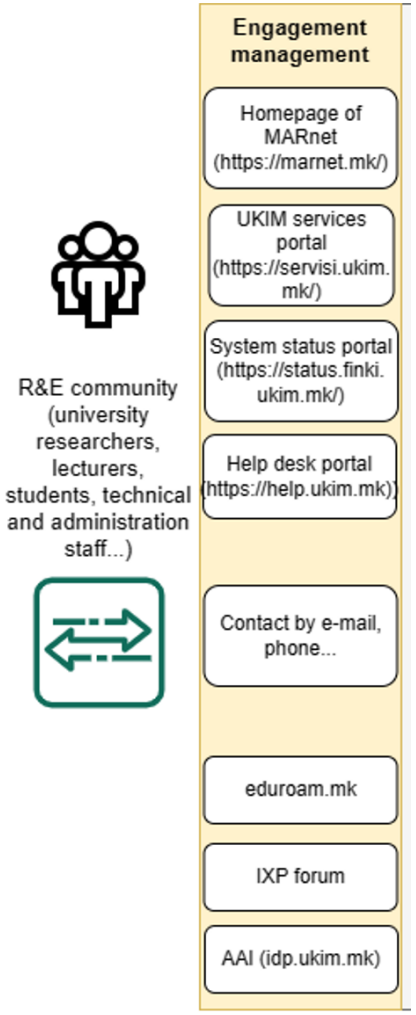
- PIONIER



- SURFnet

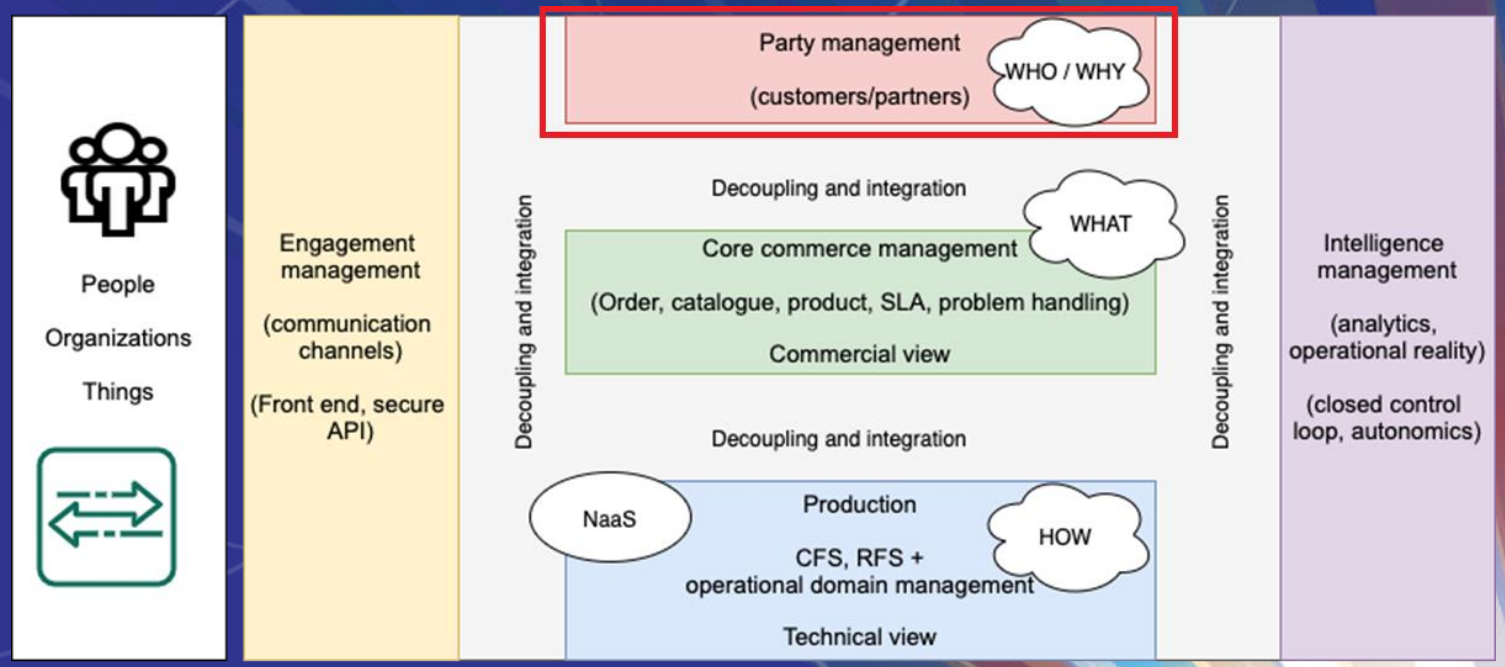


- MARnet



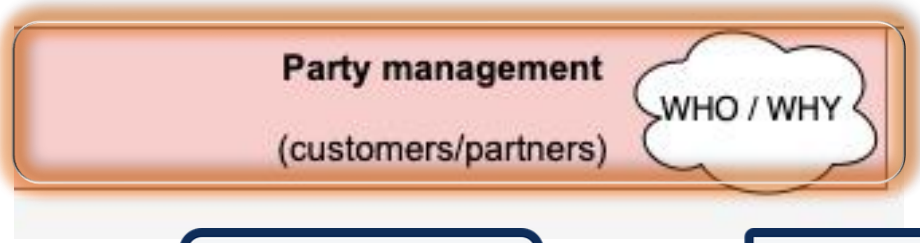
- GEANT

Party management



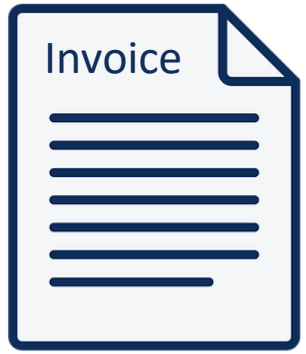
- All the information and processes related to internal and external parties.
- Parties can be customers, researchers, business partners, employees or any person or organisation related to the business

Party Management



CRM or customer relationship management tool

contract management tool



billing related to customers

Images from pixabay

Party Management Block: Functionalities

In charge of party oriented processes, functions and repositories:

- Management of party information, privacy, roles and rights
- Party interaction management
- Billing account management and bill production
- Management of financial activities with collaborating parties
- Market and external party engagement activities

Mapping of functional blocks to ODA party management architecture: business/operational processes

- It should be focused on the tactical operating model. Some examples:
  - Staff training
  - Inventory management
  - Bill invoice management
  - Brand management
- The mapping of information entities can be done in two steps:
  - Mapping of entity domains: e.g. staff, market (for example, related to students), suppliers...
  - Each entity domain can be split up into specific entities. For example:
    - Suppliers
      - Supplier party roles
      - Supplier interaction
      - Billing
      - ...

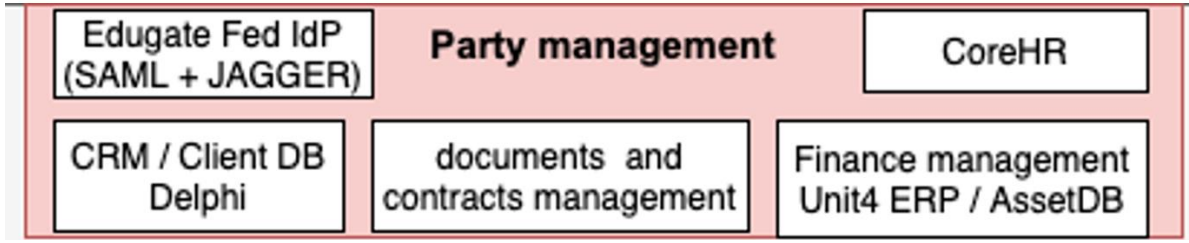
Example of Party Management

- CARNET

- CYNET

- GRNET

- HEAnet



Example of Party Management

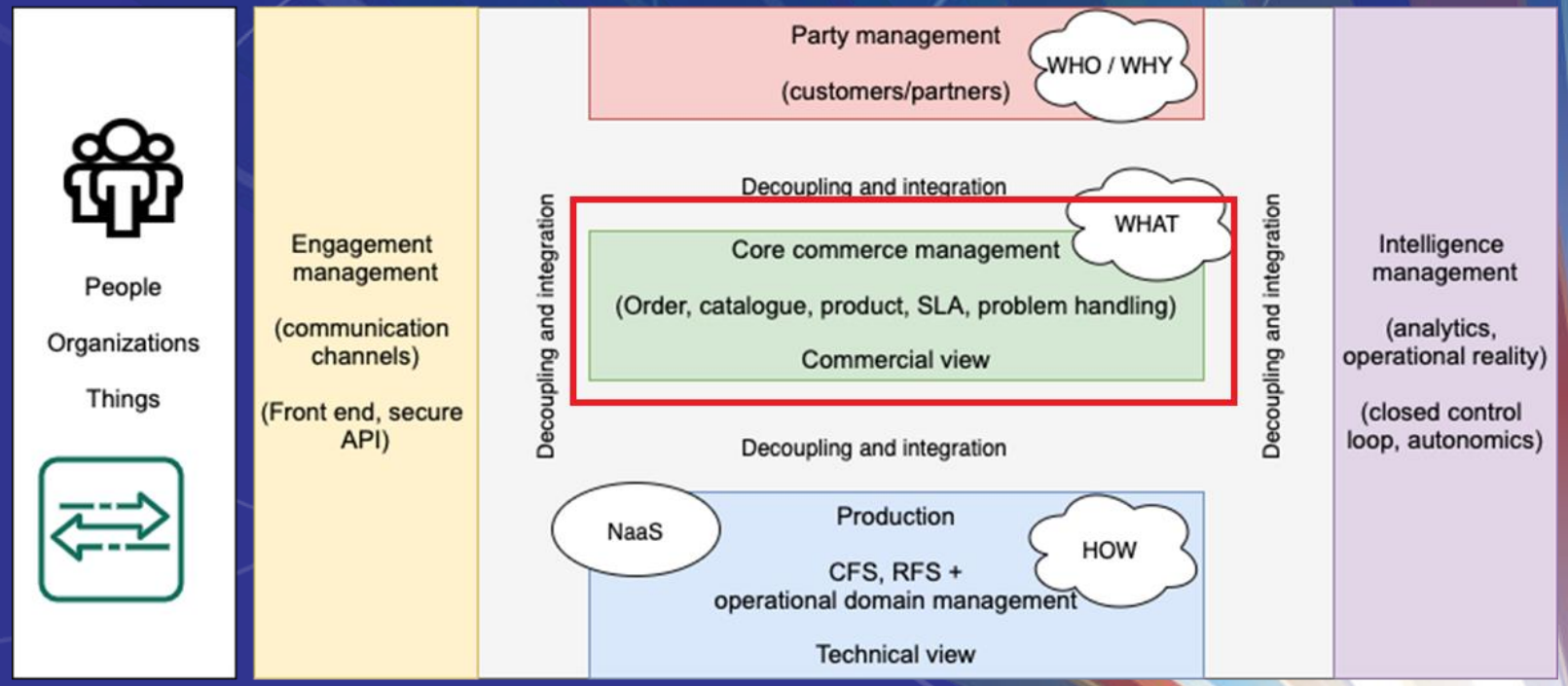
- PIONIER

- SURFnet

- MARnet

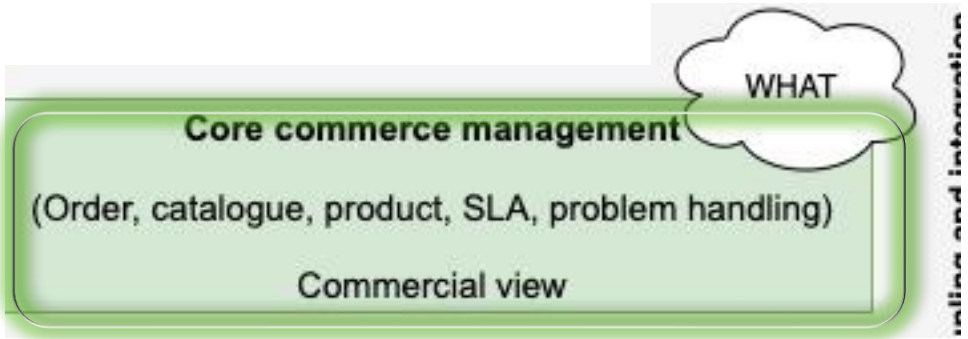
- GEANT

Core commerce management



- The exchange of goods and services
- Including the activities that facilitate them, like marketing and sales, sourcing and procurement, etc.

Core Commerce Management

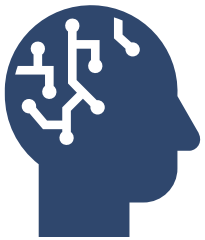


ticketing system to track ordering related tickets

Product

Inventory

bill items calculation



product strategy planning

Core Commerce Management definition

Focuses on WHAT

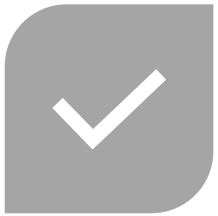
Includes all actions related to the Product Offers for all types of business engagements

Independent of the technology aspects

Product and Product Offer

Main CCM Functions

PRODUCT STRATEGY



PRODUCT OFFERS AND PRODUCT CATALOGUE MANAGEMENT

ORDER HANDLING

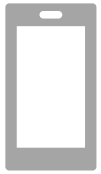
PRODUCT ASSURANCE

BILL ITEMS CALCULATION

RATING OF CHARGES

Main Interfaces Exposed by CCM

Product Catalogue Management API



Product Ordering API

Product Inventory Management API

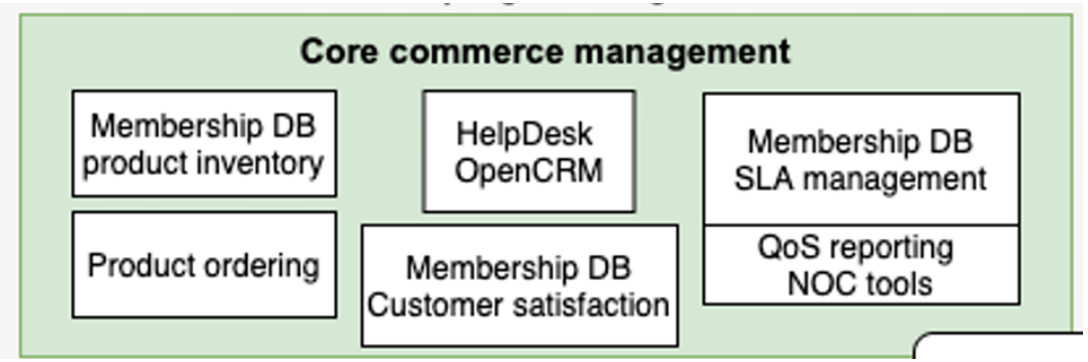
To be able to orchestrate the product lifecycle management, a number of application programming interfaces (APIs) need to be exposed

Trouble Ticket API

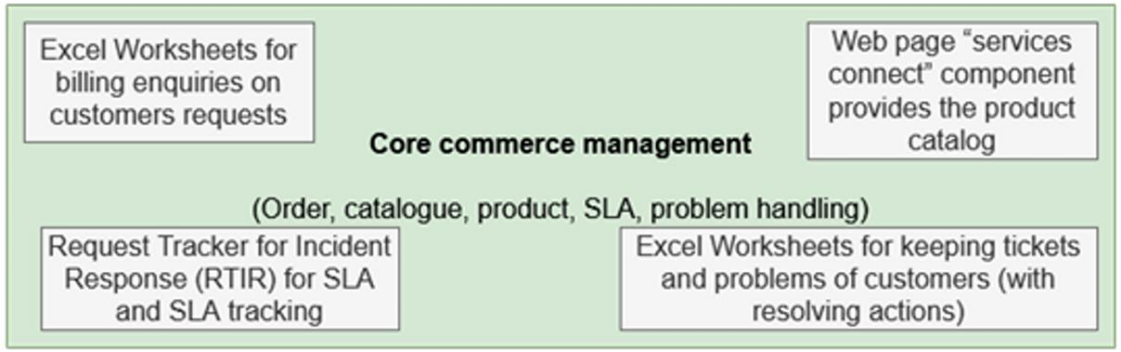
SLA Management API

Example of Core Commerce Management

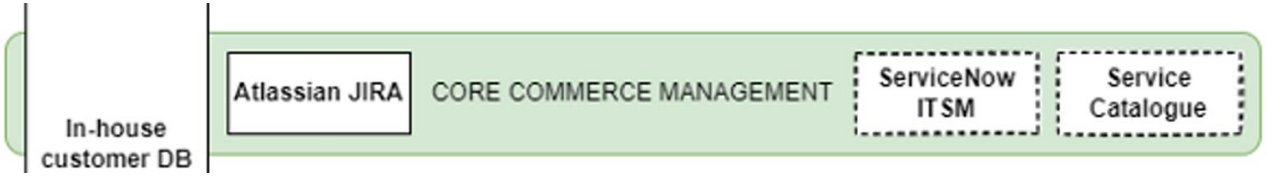
- CARNET



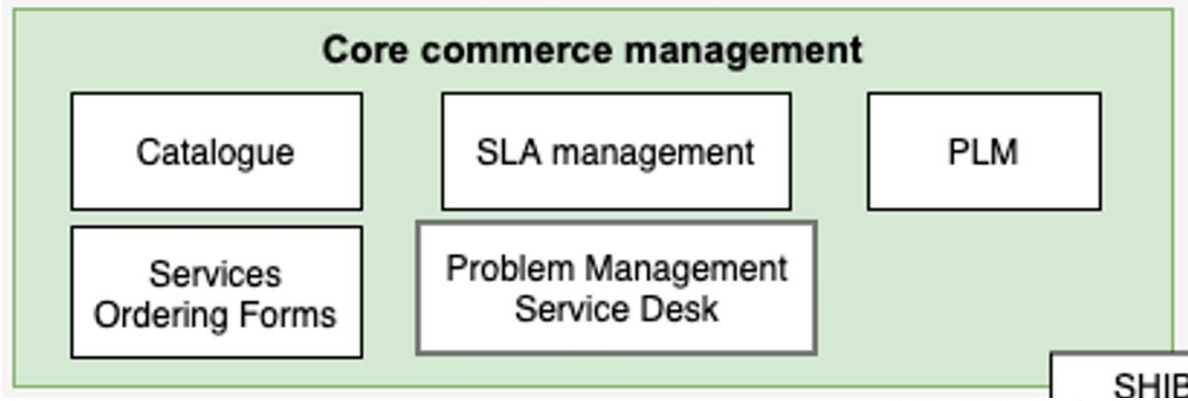
- CYNET



- GRNET

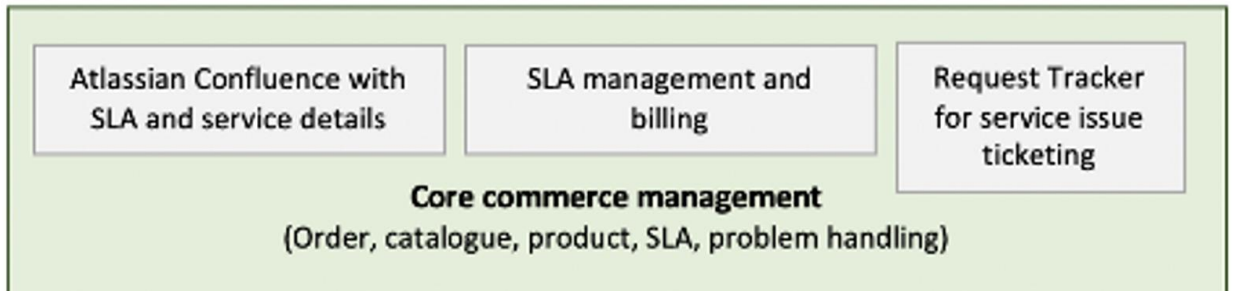


- HEAnet



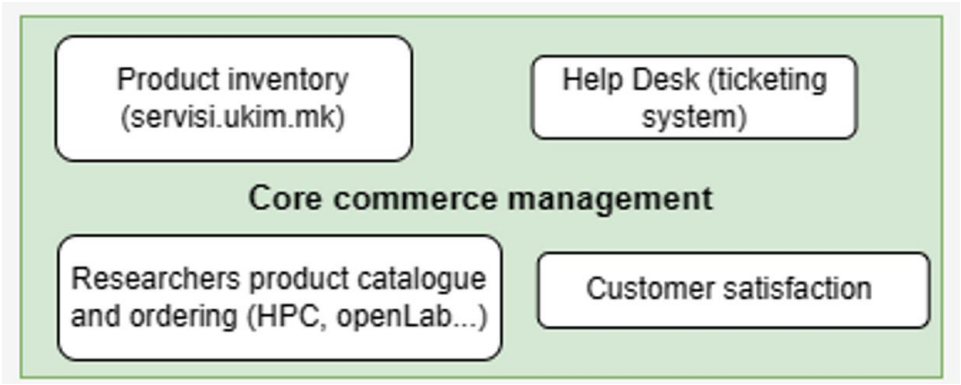
Example of Core Commerce Management

- PIONIER



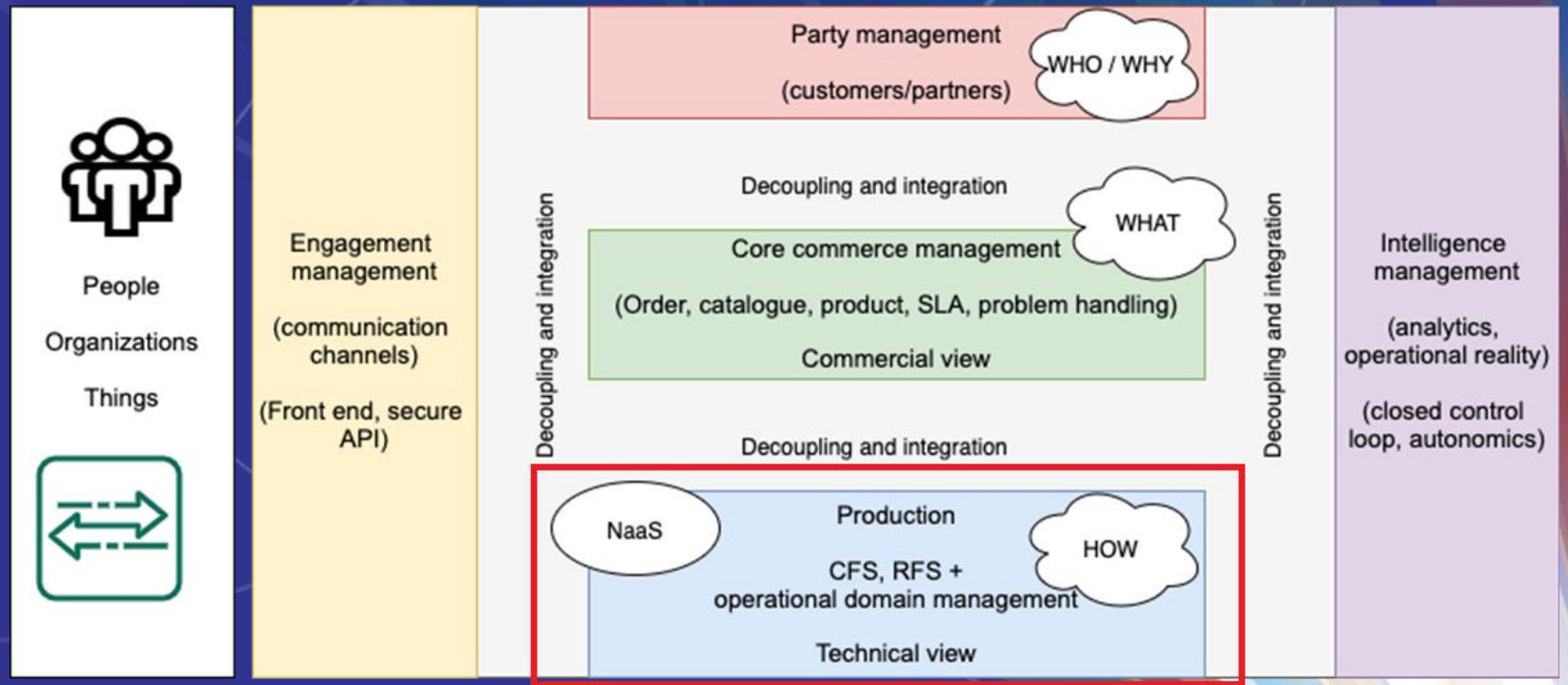
- SURFnet

- MARnet



- GEANT

Production



It is where the technical stuff happens and where most of the technical processes, tasks and tools in the NOC of a research and education institution would fall

Production

service and resource trouble tickets

Monitoring

SNMP

Automation

Programming languages

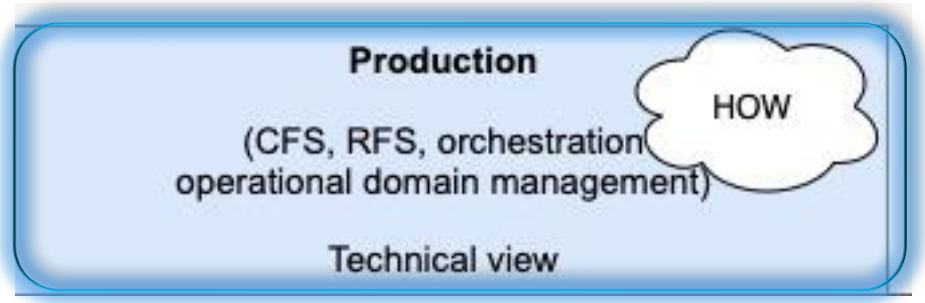
Virtualisation

IPAM tool

Resource Inventory

Service / Resource Orchestrator

Service Inventory



Network Infrastructure

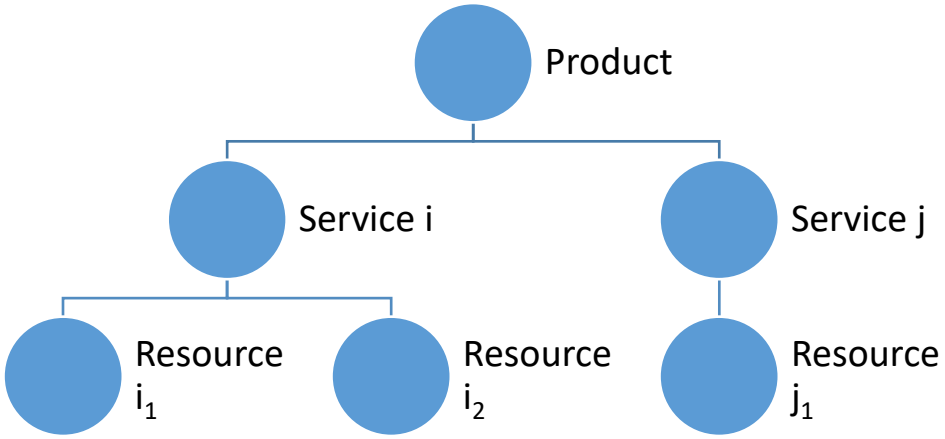
optical network

Equipment

layer 2, or layer 3 equipment

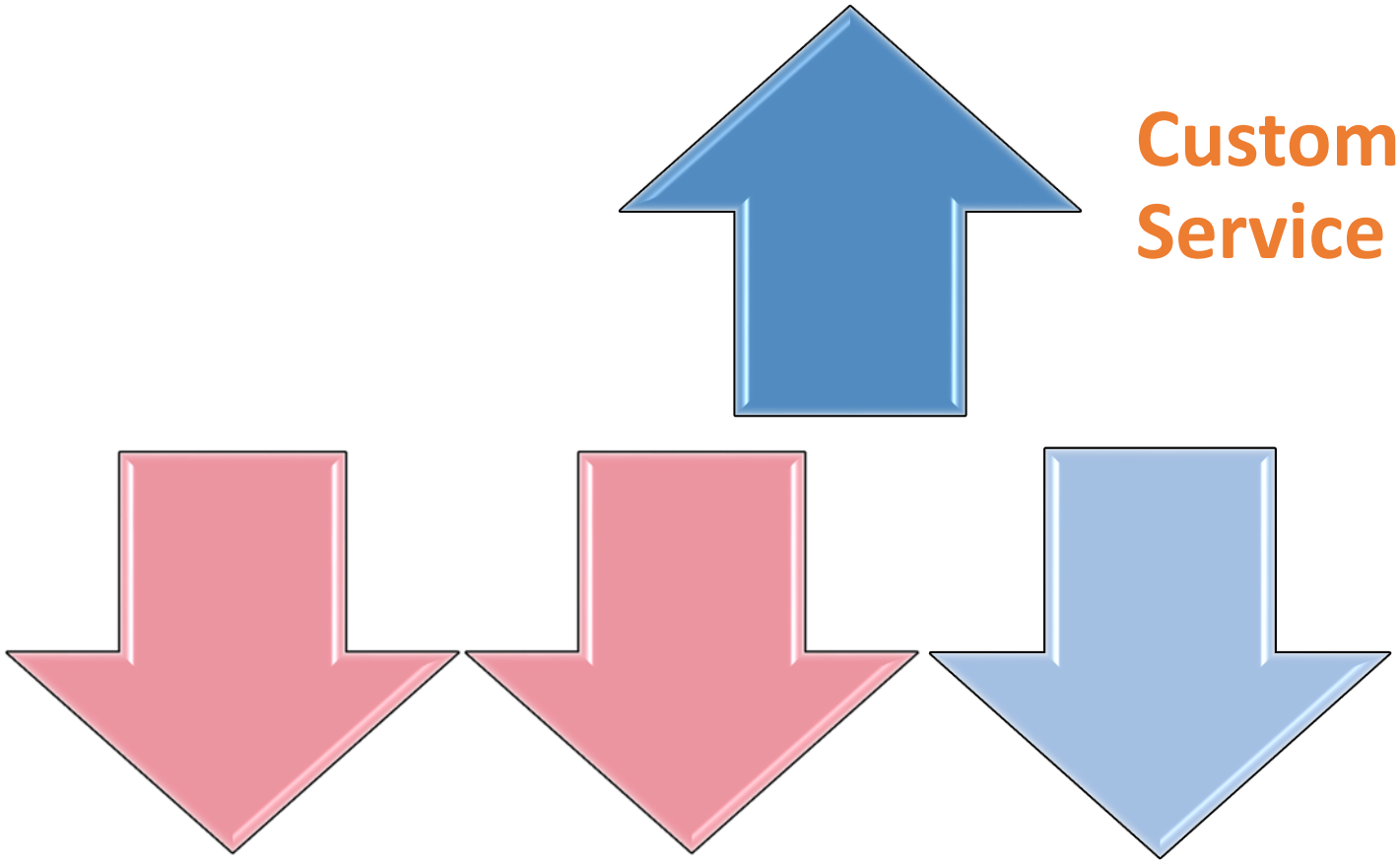
In an ideal world, this domain should expose a Network as a Service API towards the other domains

Products, Services and Resources



- The Product is the entity that is being sold by the organisation
- Each Product can be composed of one or multiple Services and Resources
- Services are implemented by the Production functional block

Products, Services and Resources (ctd.)



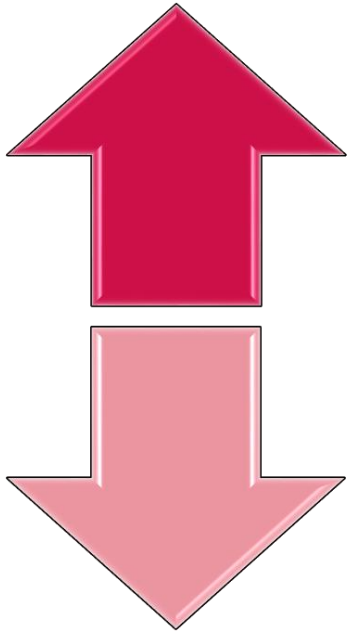
Customer-Facing Service (CFS)

Resource-Facing Service (RFS)

Physical or virtual resources or features or functions

The Main Functionalities of Production

- Responsible for End-to-End (E2E) Service Management & Orchestration
- Decouples how services are implemented from the way services are offered to the users
- Differentiates between CFS and RFS to enable transparent and flexible implementation of the requested services over different technology domains



Customer Facing Services (CFSs)

➔ technology *agnostic*

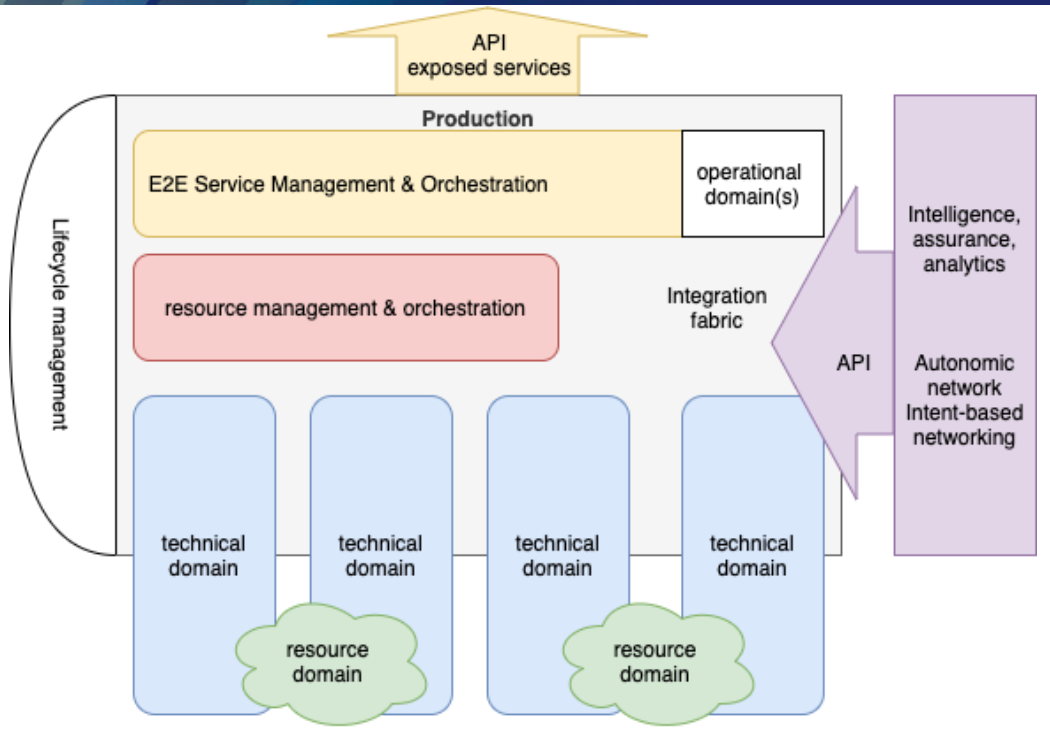
Resource Facing Services (RFSs)

➔ technology *specific*

Operational Domain Management

- Operational domain manages the complete lifecycle of all services and resources in its domain
- Defines scope of operations with an administrative boundary and a technical boundary
- Administrative boundary
  - Defines what is offered by the organisation itself and what is handled by external parties
- Technical boundary
  - Defines services and resources processed in the operational domain

Technology Management



- Production function block manages all technical domains, including different vendors and technologies
- RFSs use abstraction/virtualisation to provide a vendor- and technology-agnostic view of underlying resources
- Technical domains manage the actual implementation of RFSs
- Technical domains implement specific management interfaces for each vendor or technology
- Functionalities in the technical domains must be exposed by APIs that can be consumed by Resource Management & Orchestration and E2E Service Management & Orchestration

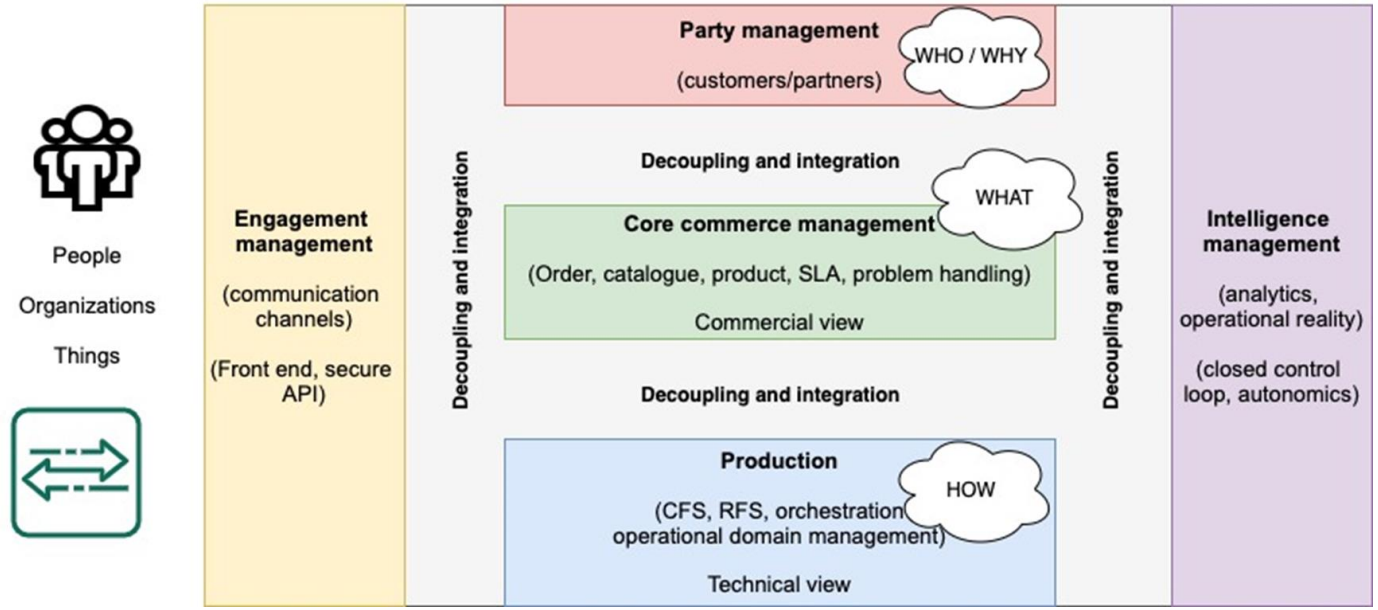
Orchestration in Production

- End-to-End (E2E) Service Management & Orchestration
  - This functionality is considered crucial in the Production functional block
  - Orchestration of service management over multiple operational domains

- Resource Management & Orchestration
  - This functionality is responsible for the implementation of the RFSs
  - Seamless integration of separate systems, processes and technologies

Automation in Production

- Automation is used to provide simplification, to make processes better, faster and more reliable with reduced or replaced human interaction
- Automation is also put in place to provide automated feedback and closed control loops



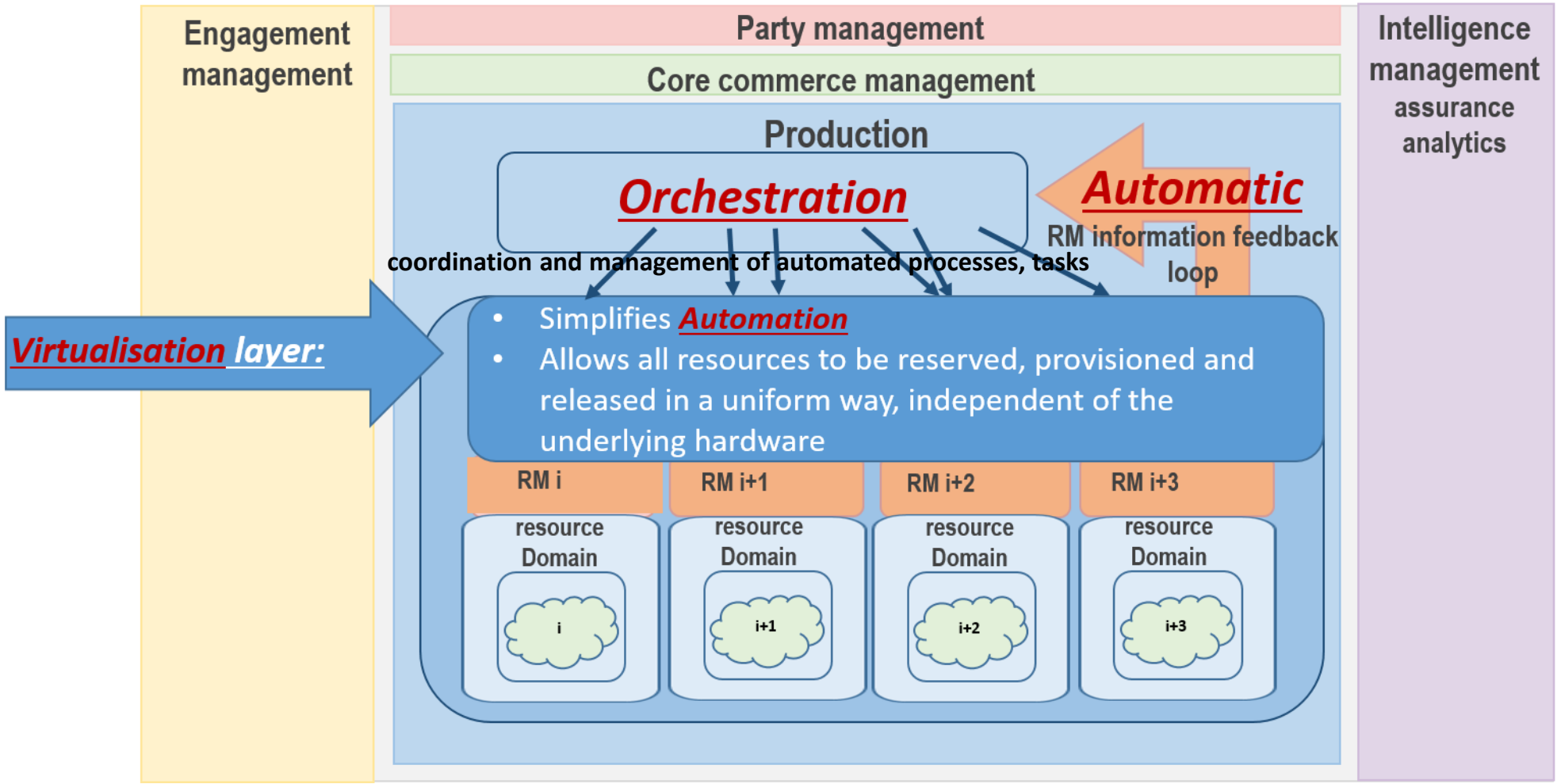
Automation and Closed Control Loops

- Different levels of automation and closed control loops in Production:
  - Via E2E Service Management and CFSs
    - Preserves the e2e service characteristics
  - Via Resource Management and RFSs
    - Technology-related service changes can be made automatically
  - Within the Technical domain
    - Very fast reaction to changes in real time or near real time.

Virtualisation of the Technical (Virtual or Physical) Resources

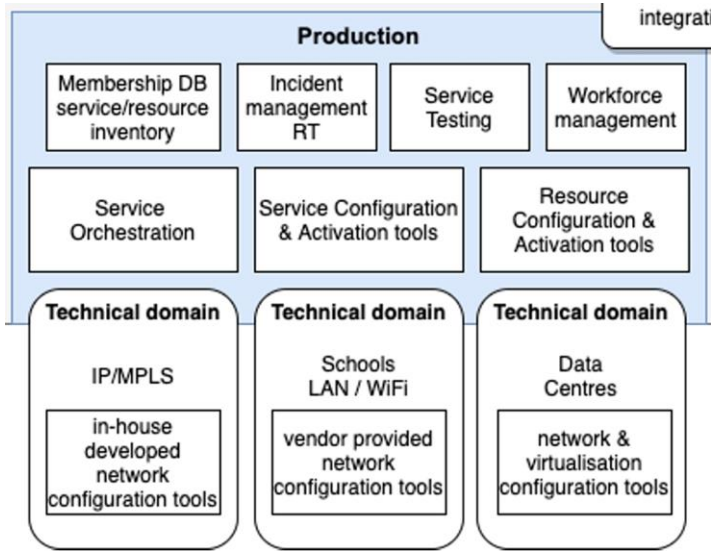
- Supports automation by improving efficiency, flexibility and dynamic response
- Automation in Production is facilitated when any new type of resource, virtual or physical, can be easily added, described and managed as all the other resources (for example, creation of VM).
- Allows the implementation of RFSs to be technology/vendor agnostic
- Network virtualisation abstracts network resources to ensure agility and technology-agnostic network service behavior

Orchestration, Virtualisation and Automation in Production

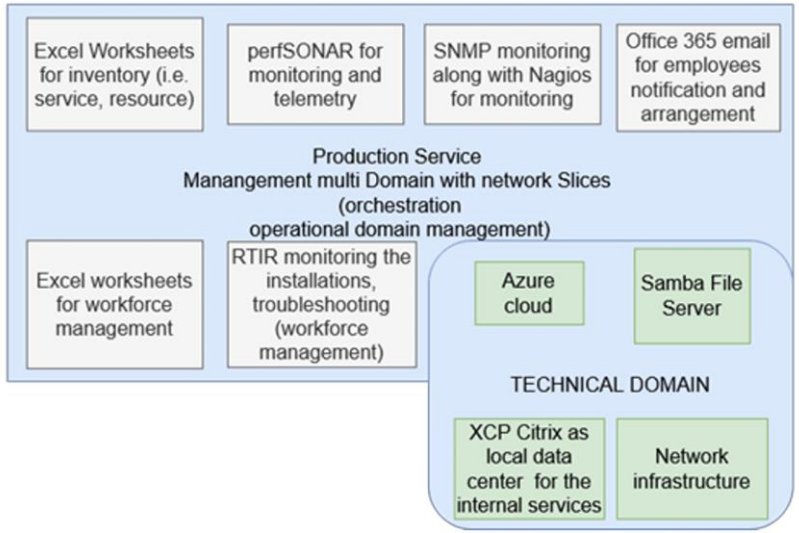


Example of Production

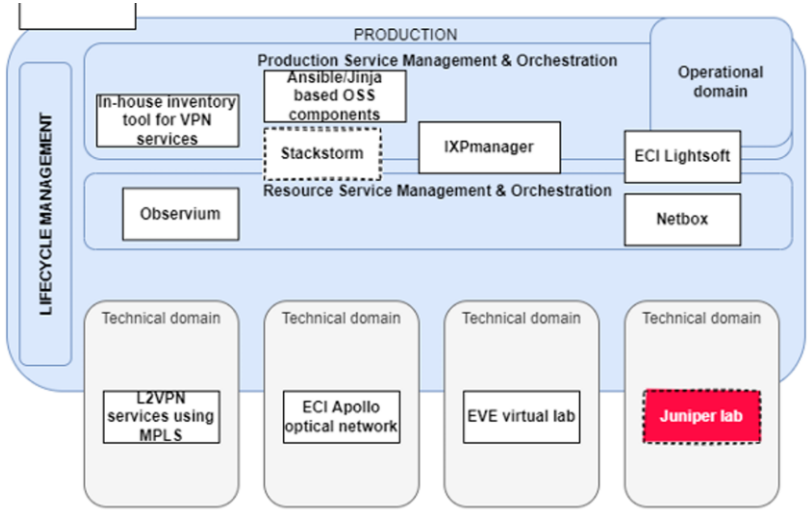
- CARNET



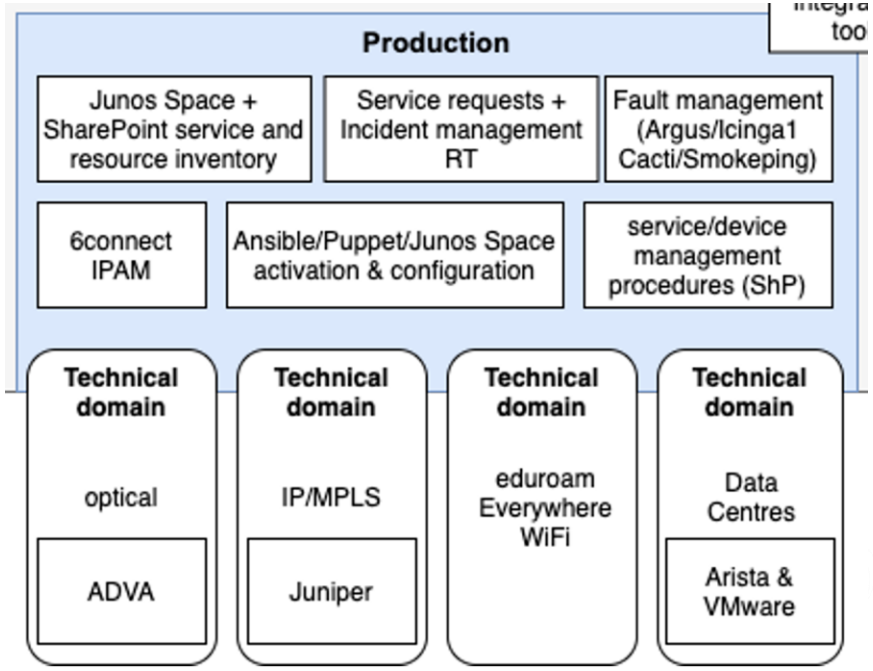
- CYNET



- GRNET

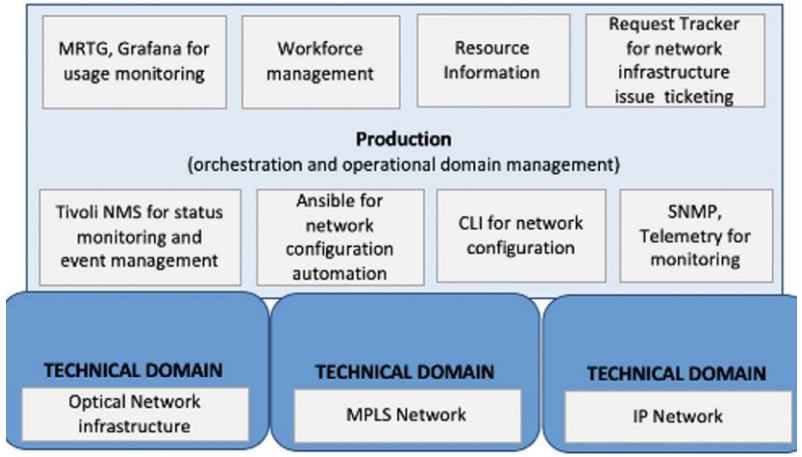


- HEAnet

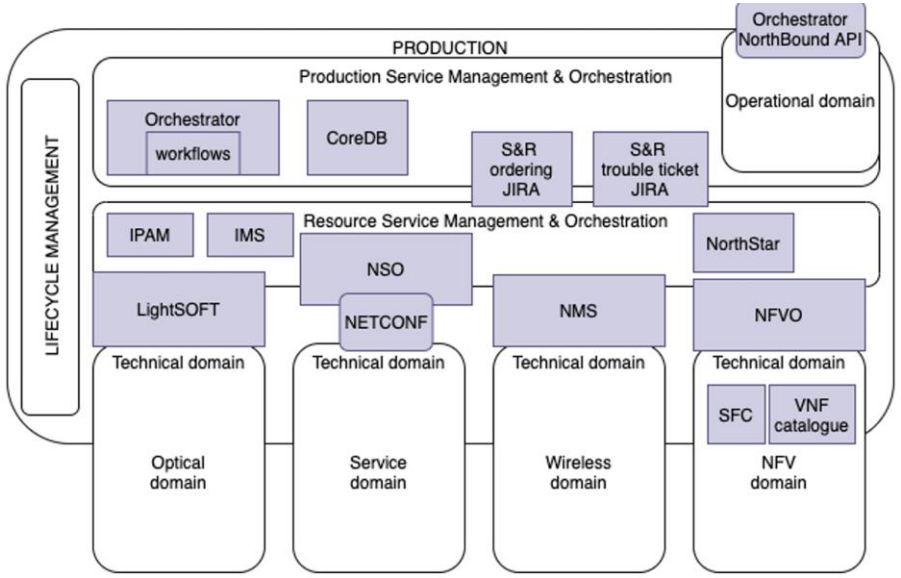


Example of Production

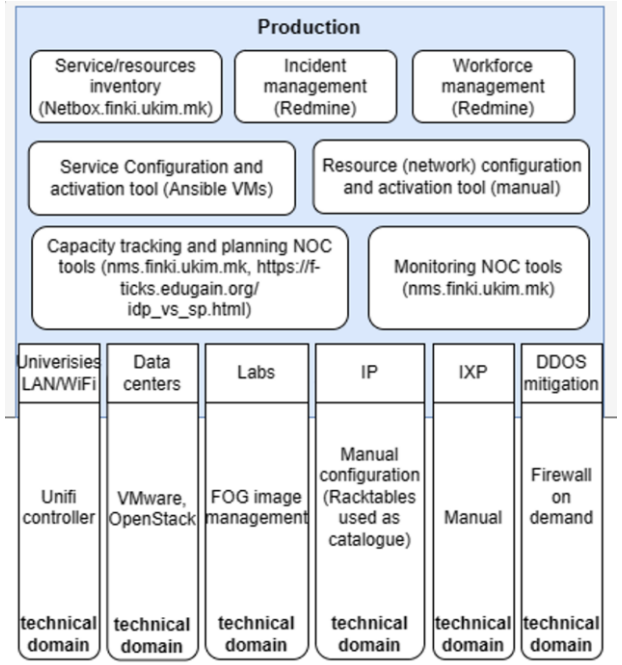
- PIONIER



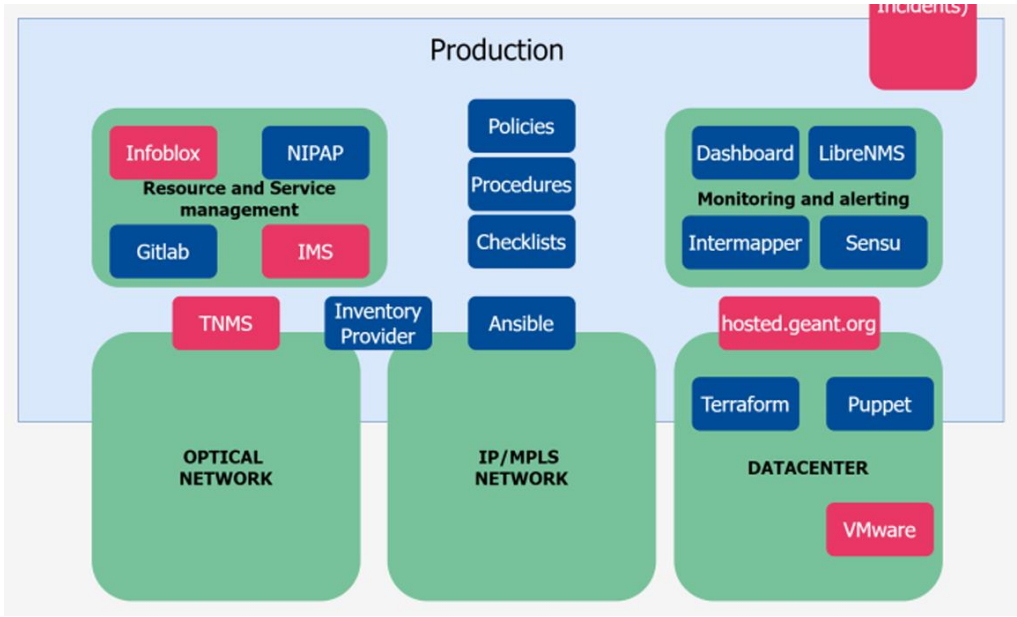
- SURFnet



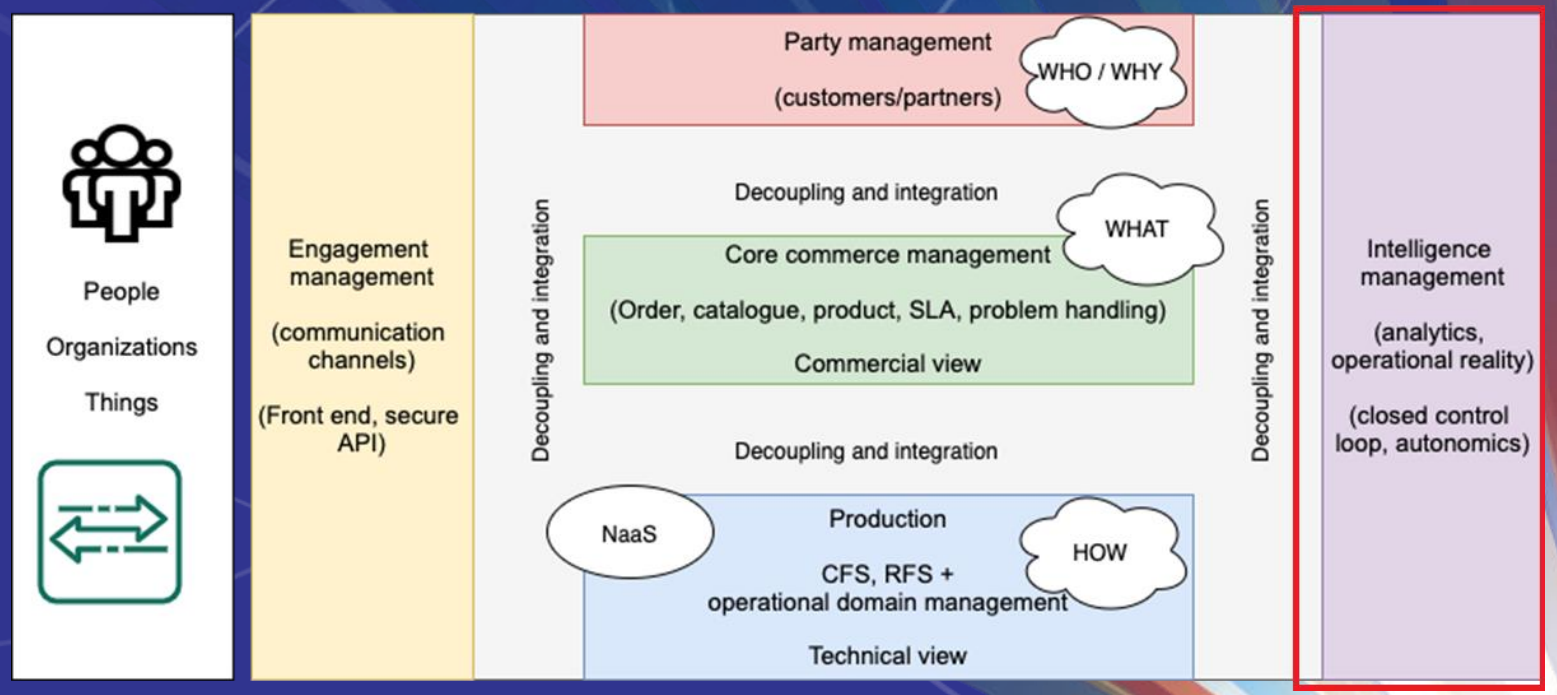
- MARnet



- GEANT



Intelligence management



- Focuses on data analytics
- Analytical processes use operational data to analyse, correlate and create their own data (for instance, KPIs)

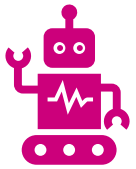
Intelligence Management

SNMP &
Telemetry

Monitoring

Artificial
Intelligence

Logs correlation



Machine

Learning

performance management, trending analysis or capacity forecasts

Intelligence Management definition

- Processes related to data analytics, using operational data from other functional blocks
- Large amounts of operational data: implementation of big-data-related capabilities is beneficial
- Data analysis techniques: e.g. trend analysis, data aggregation
  - Marketing and sales forecasting, network performance evaluation
  - Uncovering of new patterns and relations

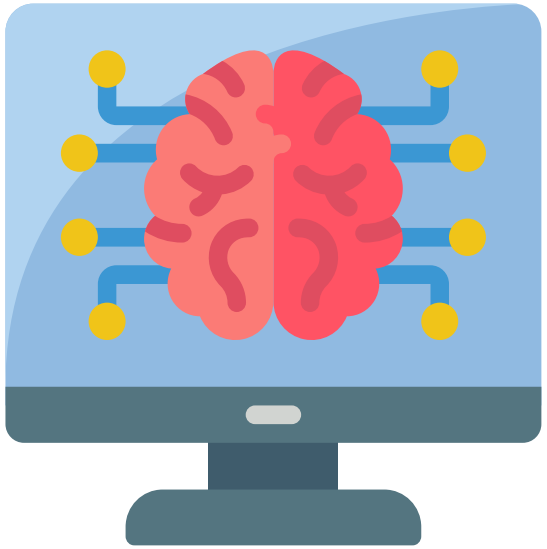
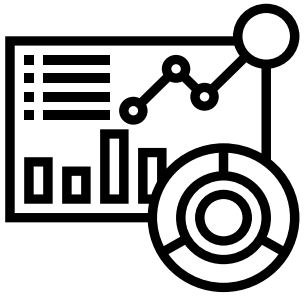
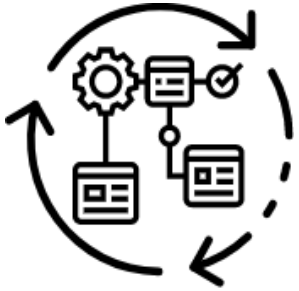


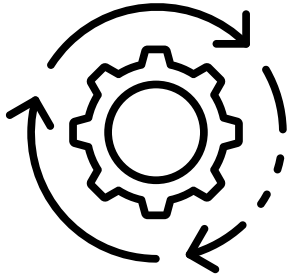
Image: www.freepik.com

Intelligence Management Block: Functionalities



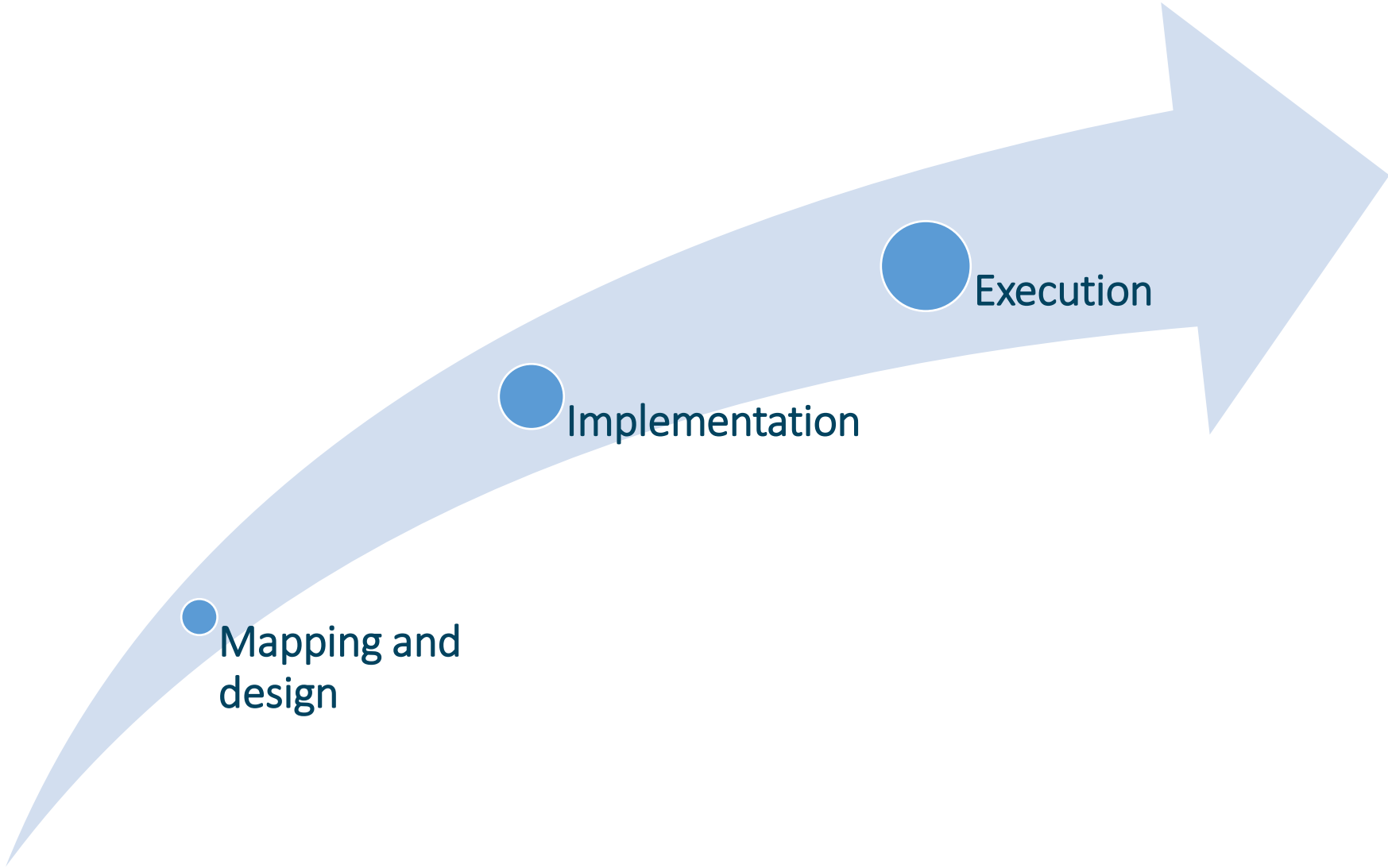
- Insight management capability
  - Uncovering of new patterns and relations based on historical data
- Autonomic manager
  - Implementation of closed-control loops
  - Knowledge management
  - Activities on different time scales
  - Implementation of self-anything (organisation, healing, tuning, etc.) capabilities





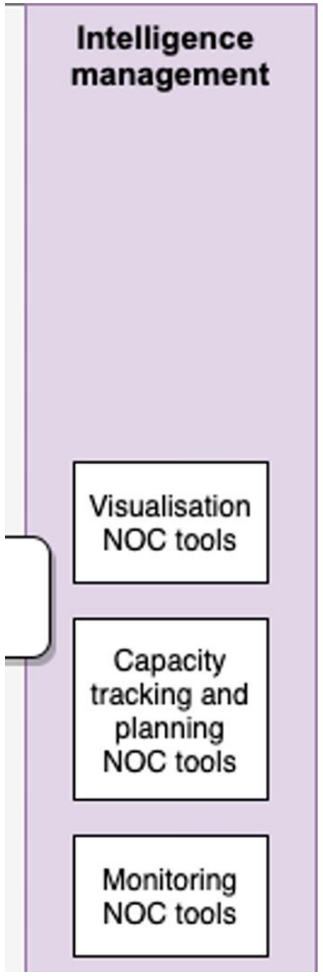
Images: www.freepik.com

TM Forum's ODA Intelligence Management Scope & Application to NRENs

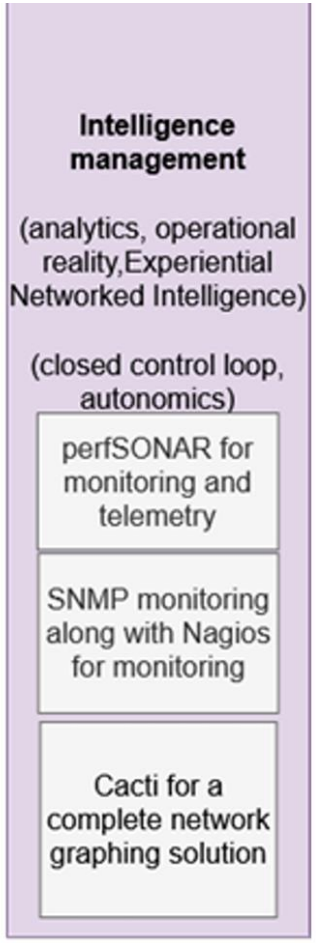


Example of Intelligence Management

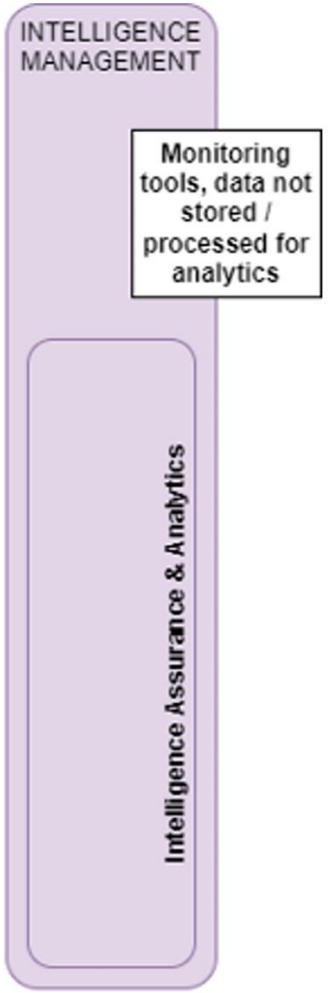
- CARNET



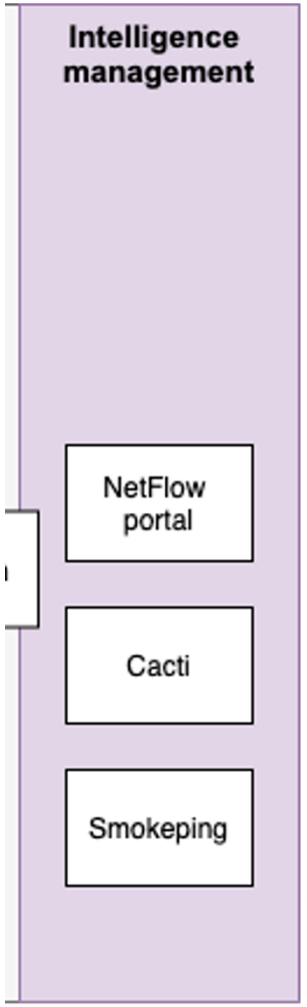
- CYNET



- GRNET

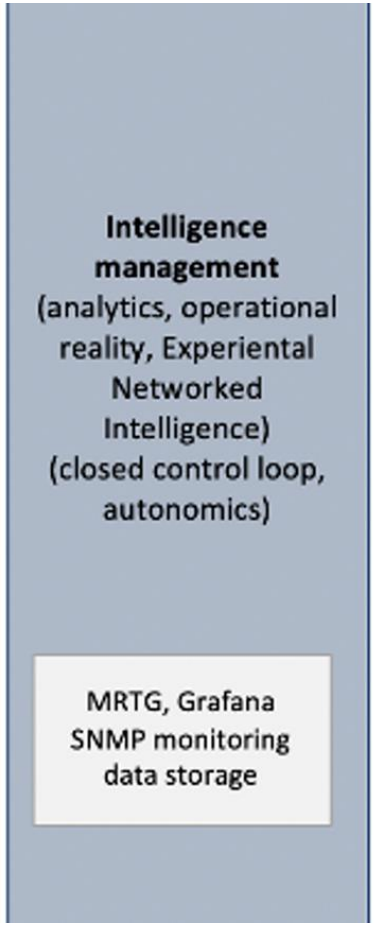


- HEAnet



Example of Intelligence Management

- PIONIER

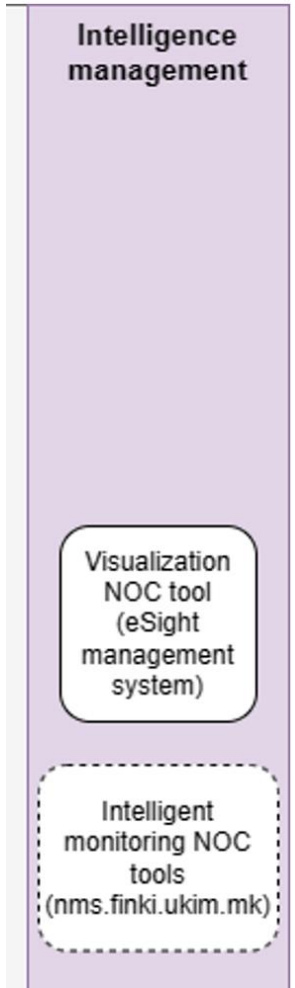


Selected datasets can be used (with the permission of PSNC management) for testing and analysis by research projects working on Artificial Intelligence/Machine Learning solutions

- SURFnet

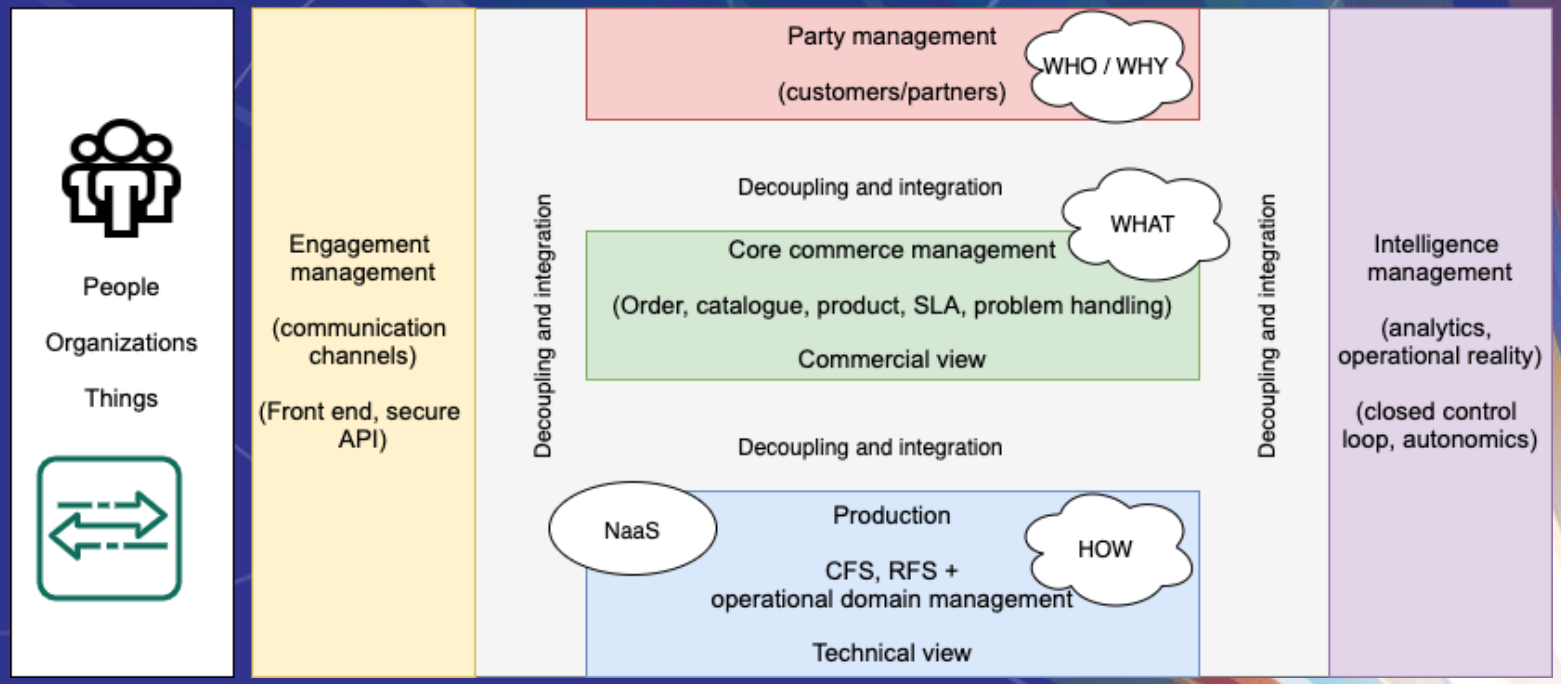
The team has also implemented some advanced data analytics using ML algorithms such as traffic prediction for capacity management

- MARnet

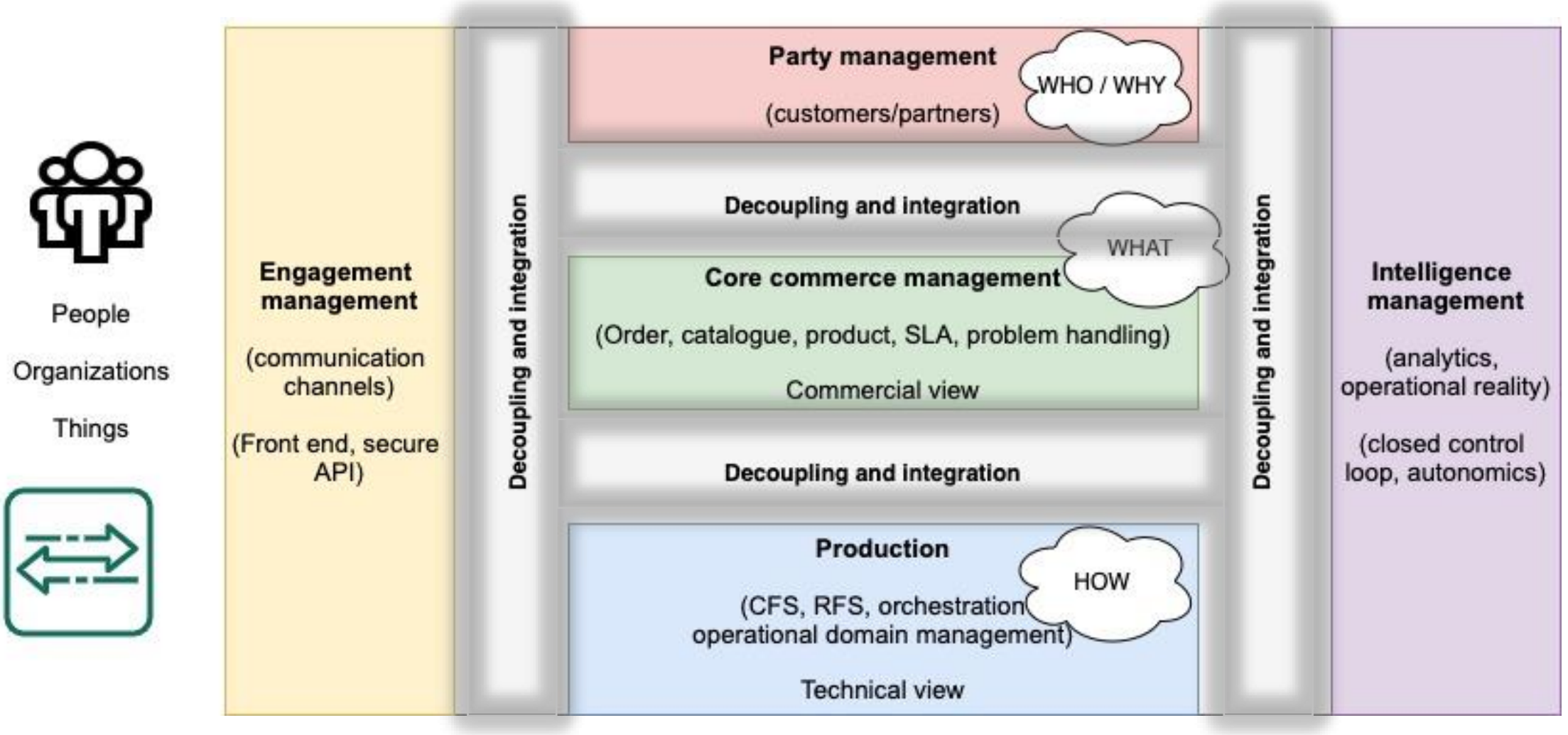


- GEANT

Decoupling and Integration



Decoupling and integration



Examples of Decoupling and integration

- The implementation of ODA's Decoupling & Integration can be associated with the manual processes and procedures that are put in place and describe how to move information from one component to another. Based on the **CYNET** NREN needs, these processes can be automated in the future.

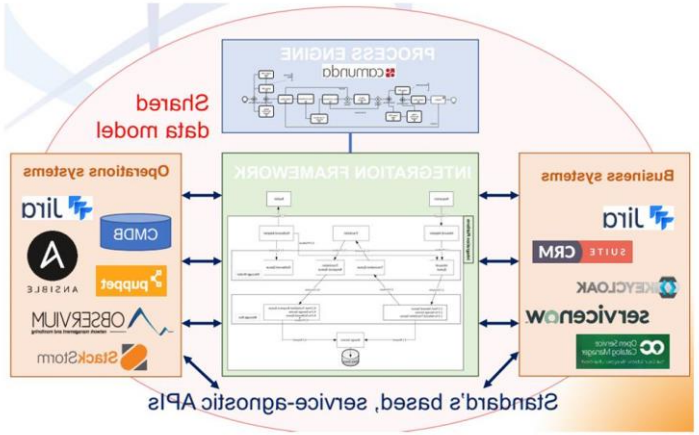
Manual processes that describe the information movement among components

- **CARNET** developed an in-house tool called Turbo Krt (Turbo Digger in Croatian) to provide a high-level view of the network, and to promote fast troubleshooting. It collects and presents all important information from different databases, and cross-compares this data with a number of active monitoring tools (Zenoss, NetFlow, syslog, network topology), and the Membership DB. It also integrates information from sources external to CARNET such as information from the power company regarding electricity outages.

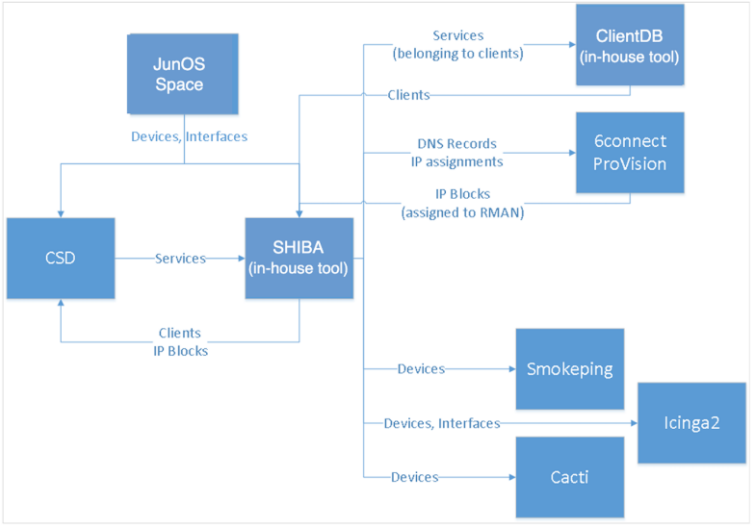
- Although a large part of the **GÉANT** workflow is based on manual processes and procedures, everything is recorded in the Ticket Management System (TTS). For example, request fulfilment is tracked from the moment the service is requested until it is deployed and operational. In addition, Jira is used to manage the work of the software development team and track feature requests and bugs.
- All communication between users and operators, and across different teams is tracked using the TTS. This is done for new deployments, updates, and incidents. From this point of view, the TTS carries out the role of decoupling and integration between all the different teams involved in the service deployment.

Examples of Decoupling and integration

- **GRNET** has prototyped (not used in production) a microservices architecture for the Integration Framework, based on the Spring Boot applications. Its building blocks are Adapter microservices, Translator microservices, Message Broker, Message Bus, and Storage microservice



- An internally developed integration tool named SHIBA (Space/**HEAnet** Integration Broker Application) takes information from both Junos Space and 6connect, and notifies Argus to start monitoring the new circuit. In addition to the automated provisioning of Icinga and Cacti, it also integrates related DNS information.



Thank You!

network-eacademy@lists.geant.org

www.geant.org

Co-funded by
the European Union