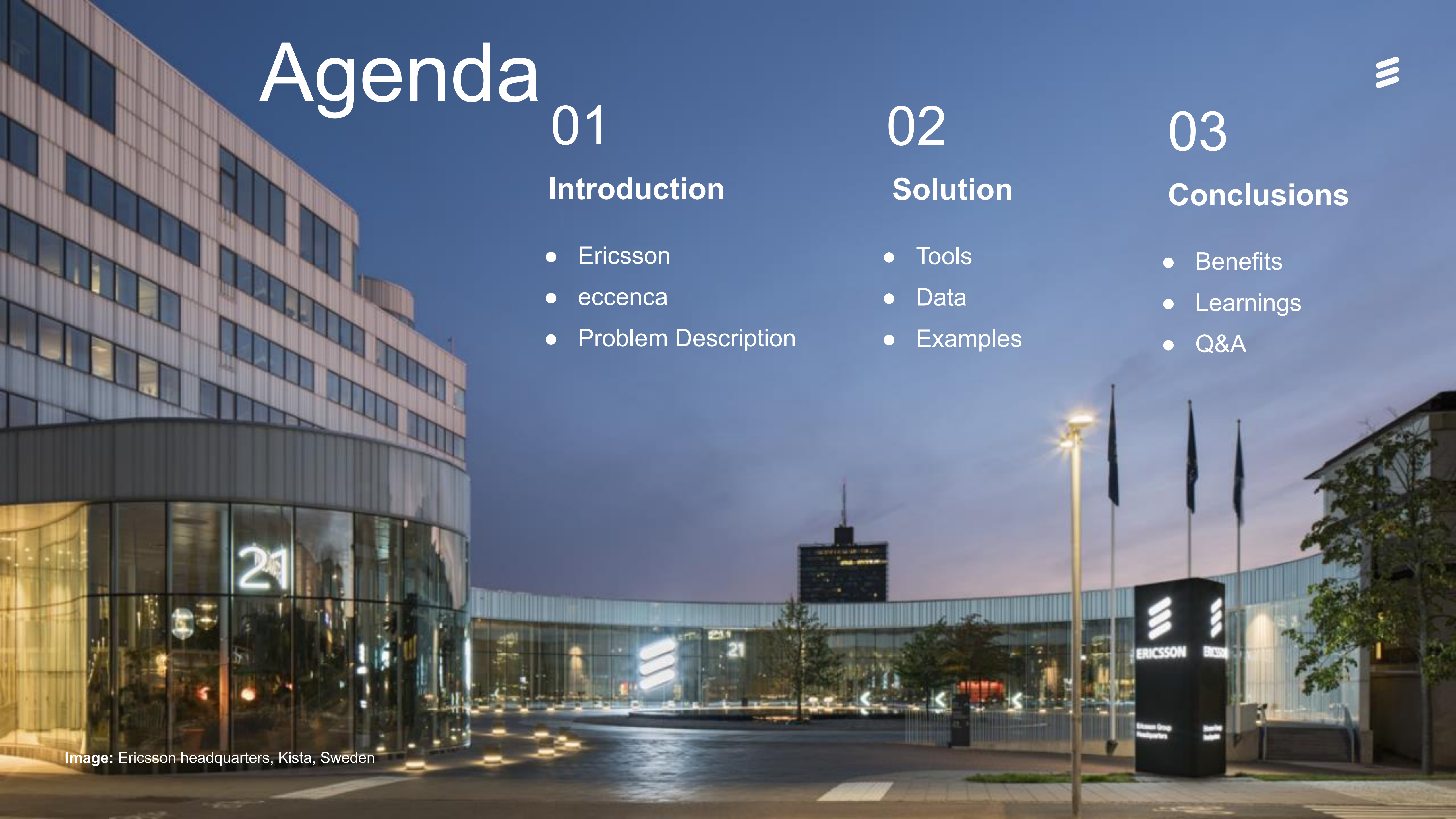
Using Knowledge Graphs to Document Ericsson Cloud Native Products

Martin Blumbach (Ericsson, Information Architect)

Marcel Fröhlich (eccenca, Director Services)

Data Week Leipzig 20

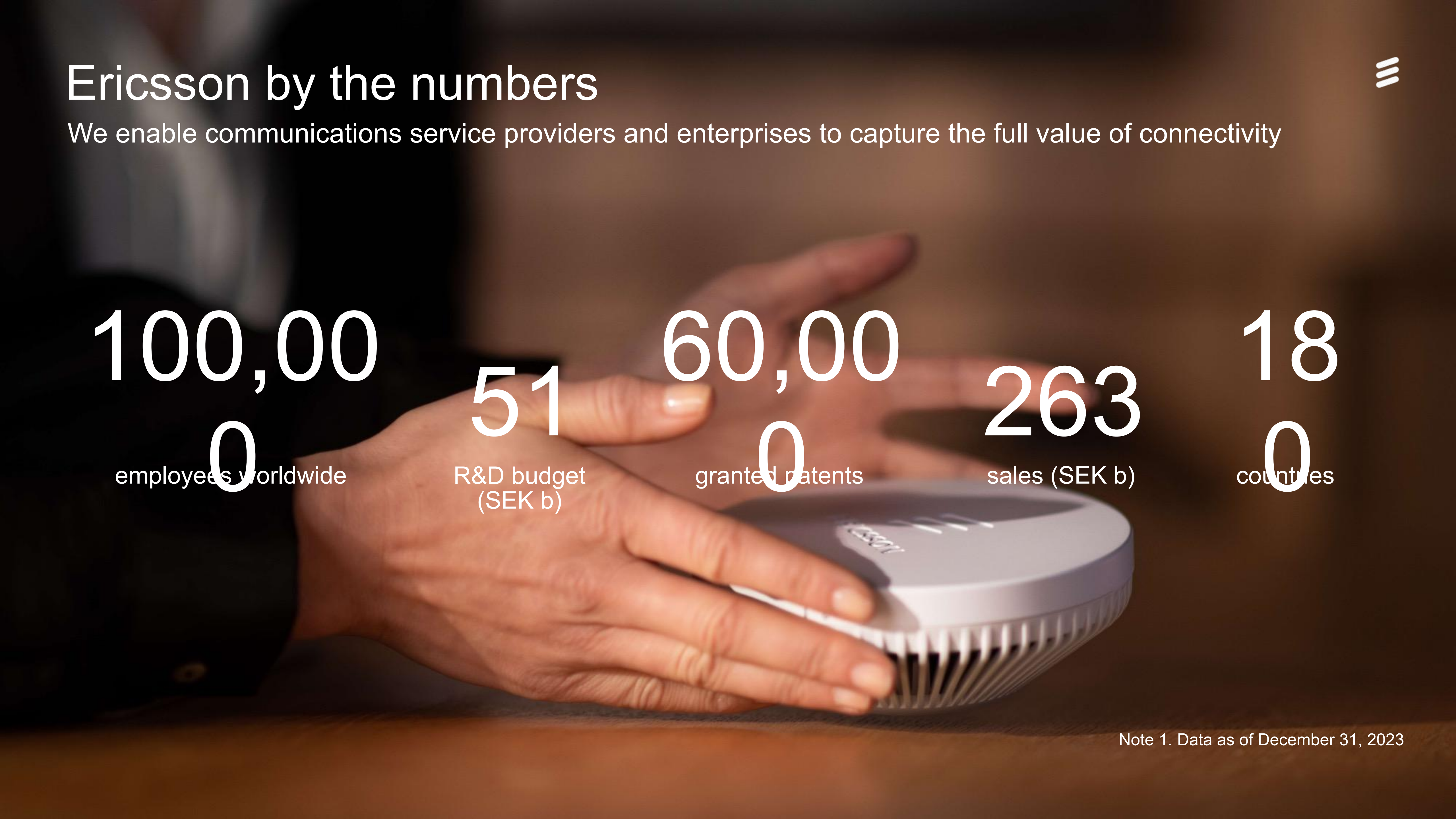


Agenda

Image: Ericsson headquarters, Kista, Sweden

Ericsson by the numbers

We enable communications service providers and enterprises to capture the full value of connectivity



100,000
employees worldwide

51
R&D budget (SEK b)

60,000
granted patents

263
sales (SEK b)

180
countries

Note 1. Data as of December 31, 2023

AIRBUS

SIEMENS

AstraZeneca

ERICSSON

TOTAL

NOKIA

NAVAL GROUP

BOSCH

DBpedia

SWM

F	A	I	R
FINDABLE	ACCESSIBLE	INTER-OPERABLE	REUSABLE & RELIABLE

SAMPLE VENDOR KNOWLEDGE GRAPHS — GARTNER® HYPE CYCLE FOR ARTIFICIAL INTELLIGENCE

COOL VENDOR — GARTNER® INTELLIGENT SUPPLY CHAIN EXECUTION TECHNOLOGIES

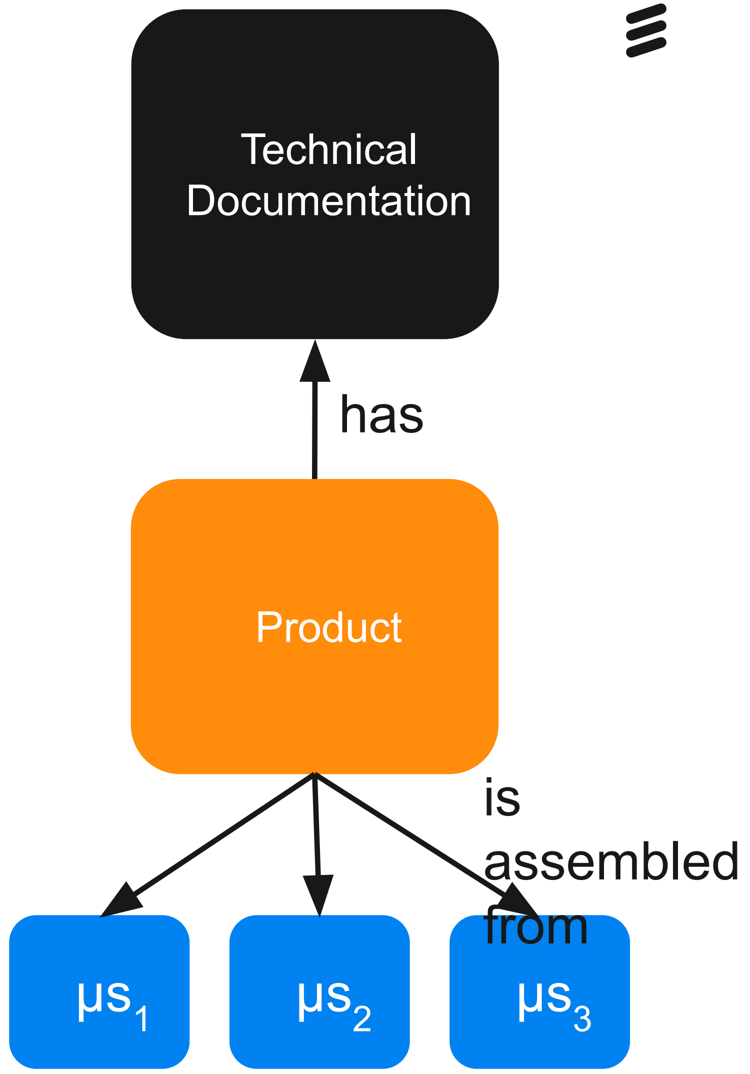
SAMPLE VENDOR DIGITAL SUPPLY CHAIN TWINS — GARTNER® HYPE CYCLE FOR SUPPLY CHAIN PLANNING TECHNOLOGIES

SAMPLE VENDOR DIGITAL SUPPLY CHAIN TWINS — GARTNER® HYPE CYCLE FOR SUPPLY CHAIN STRATEGY

WINNER — futureSAX INNOVATION AWARD

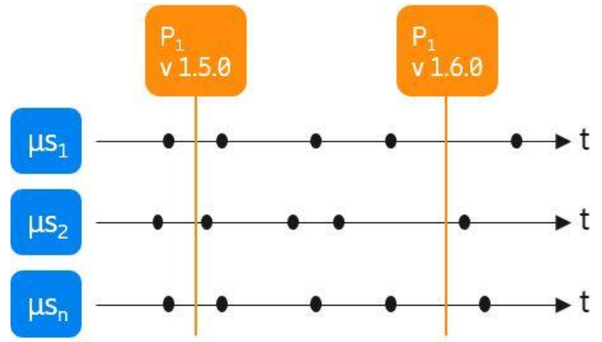
eccenca
mastering complexity

How to create technical documentation for (software) products flexibly assembled from microservices independently and frequently released?

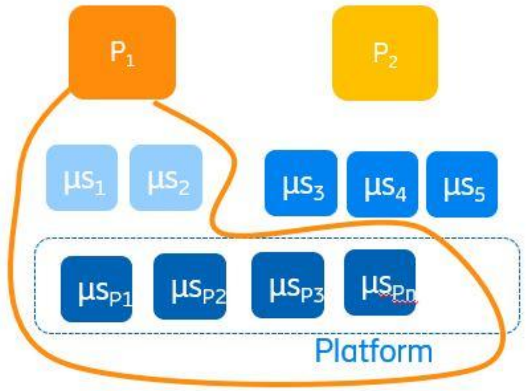


Cloud Native: “Flexibly”, “Independently”, “Frequently”

Development Manager
“I want microservices independently life cycled with high release frequency.”
Tech Docs
“Uh. Oh. Herding cats.”

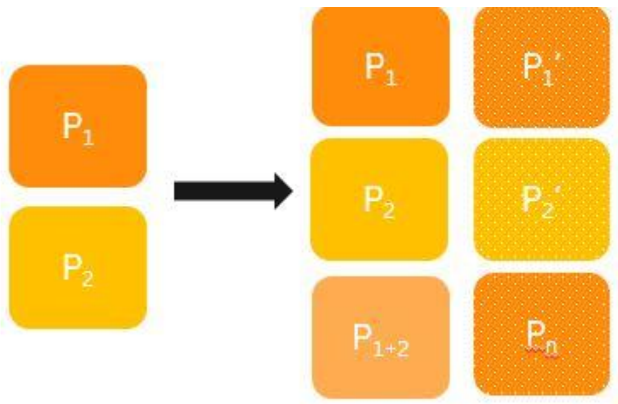


Development Manager
“Use common microservices for development efficiency.”
Tech docs
“We need a non-trivial enterprise content reuse strategy.”



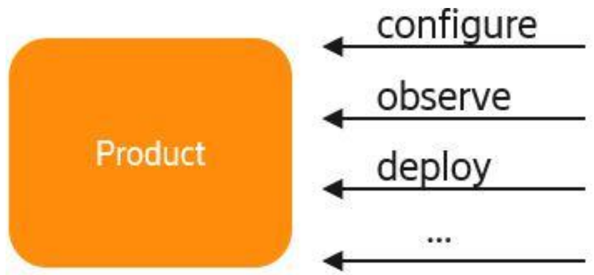
Product Manager
“I can sell more product variants and get features out faster.”

Tech Docs
“We need far more tech writers and more SME support.”



Product User
“I don’t care about microservices!”

Tech Docs
“We thought we could create one document per microservice. Sorry. **Will do better.”**



Requirement: Manage all content per microservice and automatically assemble it from the perspective of a product user

~~**Solution A**~~

~~Use documents per microservice~~

Naïve

~~**Solution B**~~

~~Use only GenAI~~

Complete and consistent technical documentation requires ground truth

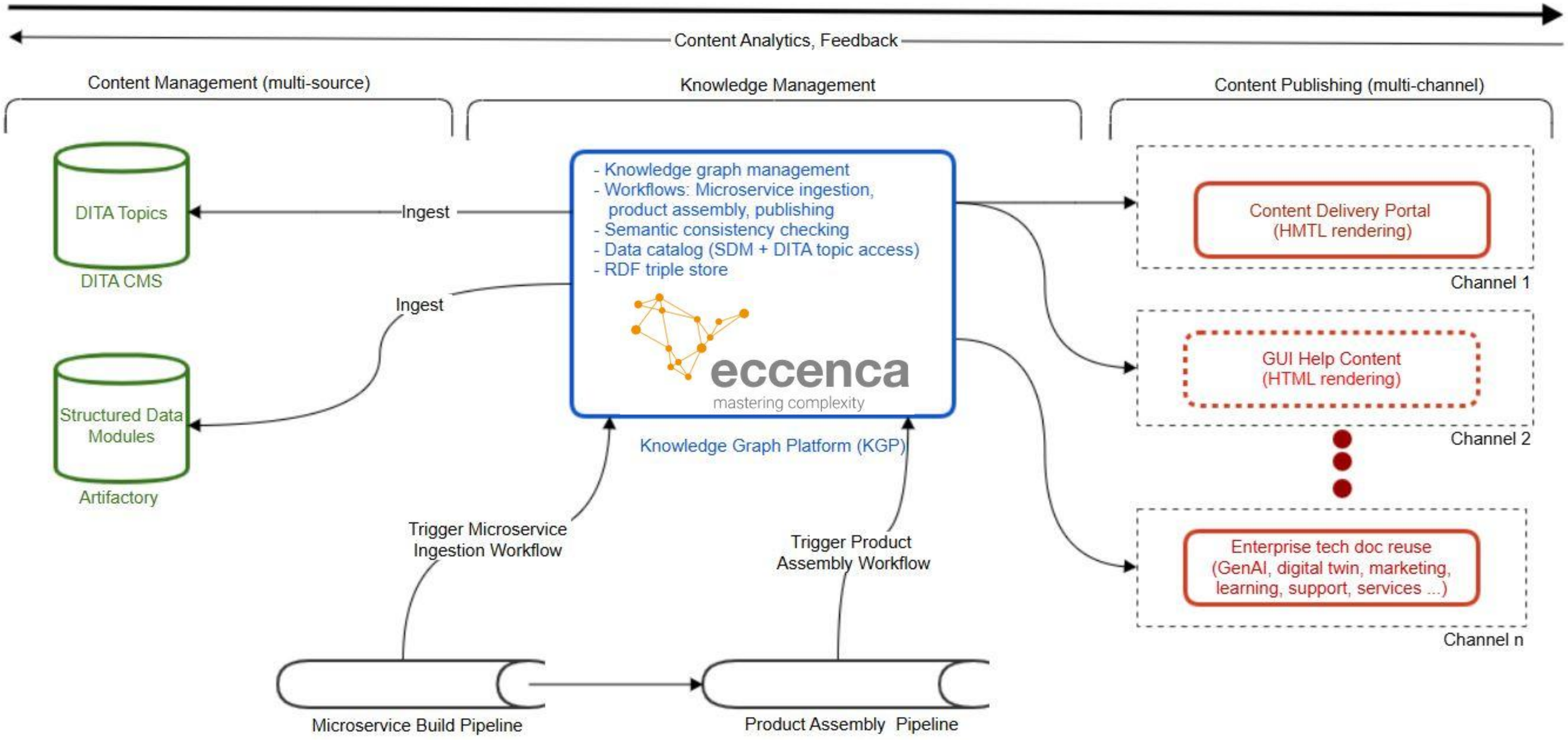
Solution C

Automate using knowledge graphs

Possible Next Step

Orchestrated automation and ad-hoc interaction with KGs using GenAI (MCP)

Automated Content Delivery Pipeline



Content Analytics, Feedback
Content Management (multi-source)
Knowledge Management
Content Publishing (multi-channel)
DITA Topics
DITA CMS
Ingest
Ingest
Structured Data Modules
Artifactory
- Knowledge graph management
- Workflows: Microservice ingestion, product assembly, publishing
- Semantic consistency checking
- Data catalog (SDM + DITA topic access)
- RDF triple store
eccenca
mastering complexity
Knowledge Graph Platform (KGP)
Trigger Microservice Ingestion Workflow
Trigger Product Assembly Workflow
Microservice Build Pipeline
Product Assembly Pipeline
Content Delivery Portal (HMTL rendering)
Channel 1
GUI Help Content (HTML rendering)
Channel 2
Enterprise tech doc reuse (GenAI, digital twin, marketing, learning, support, services ...)
Channel n

Variants Complexity and Quantities

- 100s of complementary microservices that can be assembled – Lego like
- 10.000s of configuration parameters

- 10s of assembly structures for software products with up to 10.000s of topics
- Central topic repository with > 100.000 topics
- Daily reconfigurations and changes

- Up to 100 documents per product and release
- Manually creating up to date documentation is impossible
- Out of the box documentation automation tools take wrong assumptions and are too simplistic

Create and Update Content **per Microservice**

DITA Topic

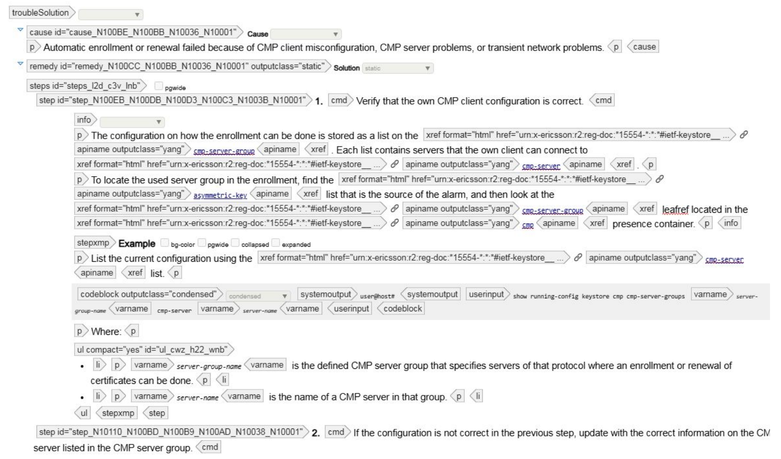
- Written "by human beings for human beings“
- Topics types: concept, task, troubleshooting, ref
- Topic “classes”: Detailed topic templates
- High semantical expressiveness
- Represented in KG by topic metadata
  - Machine-readable facts

Findable Accessible Interoperable Reusable

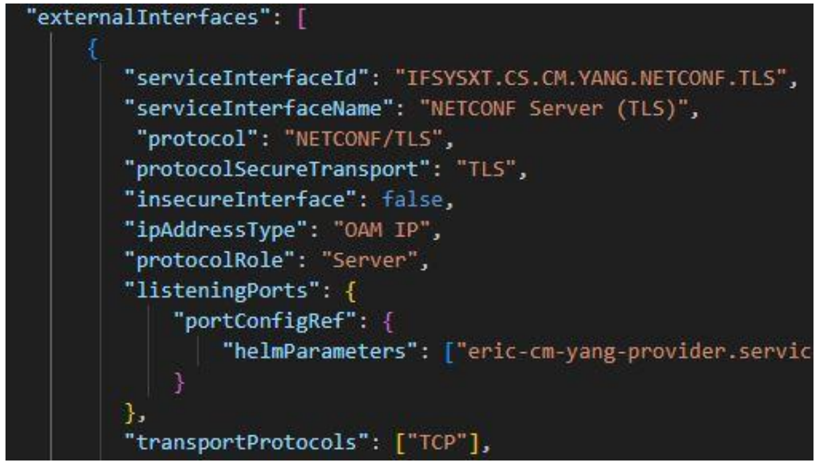
Icon license

- Reuse development artifacts (“config files”)
- Follow FAIR data principles
- JSON, XML incl. versioned schema
- Used to instantiate the knowledge graph

Structured Data Module



```
troubleSolution
cause id="cause_N100BE_N100BB_N10036_N10001" Cause
p Automatic enrollment or renewal failed because of CMP client misconfiguration, CMP server problems, or transient network problems. p cause
remedy id="remedy_N100CC_N100BB_N10036_N10001" outputclass="static" Solution static
steps id="steps_l2d_c3v_lnb" pgwide
step id="step_N100EB_N100DB_N100D3_N100C3_N1003B_N10001" 1. cmd Verify that the own CMP client configuration is correct. cmd
info
p The configuration on how the enrollment can be done is stored as a list on the xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ...
apiname outputclass="yang" cmp-server-group apiname xref . Each list contains servers that the own client can connect to
xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ... apiname outputclass="yang" cmp-server apiname xref . p
p To locate the used server group in the enrollment, find the xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ...
apiname outputclass="yang" asymmetric-key apiname xref list that is the source of the alarm, and then look at the
xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ... apiname outputclass="yang" cmp-server-group apiname xref leafref located in the
xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ... apiname outputclass="yang" cmp apiname xref presence container. p info
stepxmp Example bg-color pgwide collapsed expanded
p List the current configuration using the xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ... apiname outputclass="yang" cmp-server
apiname xref list. p
codeblock outputclass="condensed" condensed systemoutput user@host# systemoutput userinput show running-config keystore cmp cmp-server-groups varname server-
group-name varname cmp-server varname server-name varname userinput codeblock
p Where: p
ul compact="yes" id="ul_cwz_h22_wnb"
• li p varname server-group-name varname is the defined CMP server group that specifies servers of that protocol where an enrollment or renewal of
  certificates can be done. p li
• li p varname server-name varname is the name of a CMP server in that group. p li
ul stepxmp step
step id="step_N10110_N100BD_N100B9_N100AD_N10038_N10001" 2. cmd If the configuration is not correct in the previous step, update with the correct information on the CM
server listed in the CMP server group. cmd
```



```
"externalInterfaces": [
    {
        "serviceInterfaceId": "IFSYSXT.CS.CM.YANG.NETCONF.TLS",
        "serviceInterfaceName": "NETCONF Server (TLS)",
        "protocol": "NETCONF/TLS",
        "protocolSecureTransport": "TLS",
        "insecureInterface": false,
        "ipAddressType": "OAM IP",
        "protocolRole": "Server",
        "listeningPorts": {
            "portConfigRef": {
                "helmParameters": ["eric-cm-yang-provider.servic
            }
        },
        "transportProtocols": ["TCP"],
```

Ontologies: Fusing Data with Knowledge

What is in the ontology:

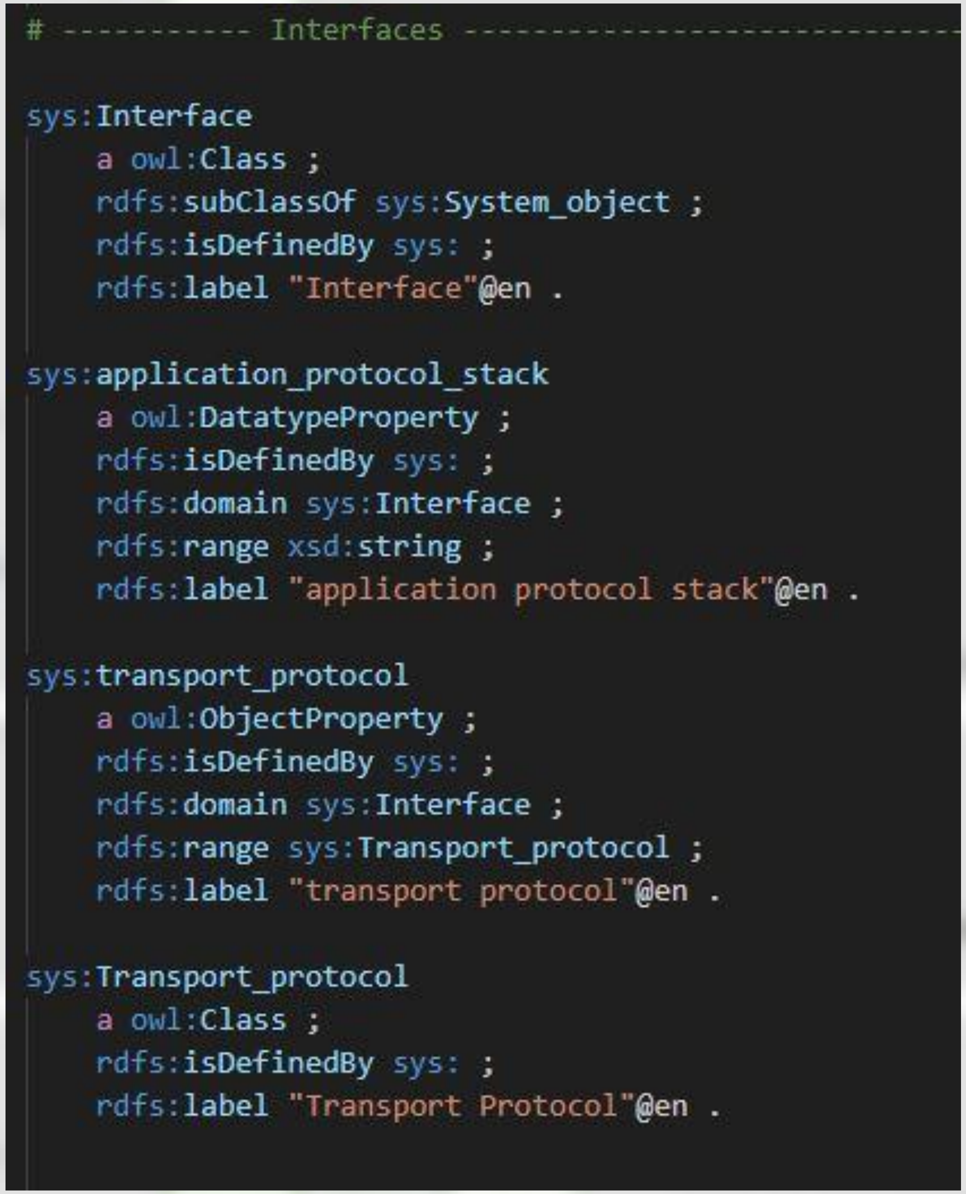
- **Product**: Products, microservices, and whatever the product user can interact with (interfaces, alarms, …)
- **Technical Documentation**: Metadata of DITA topics, DITA topic hierarchies, publishing system related entities …
- **Relations and properties** of the above

What is **not** in the ontology:

- Everything involved people know but is not relevant
- Context specific data constraints

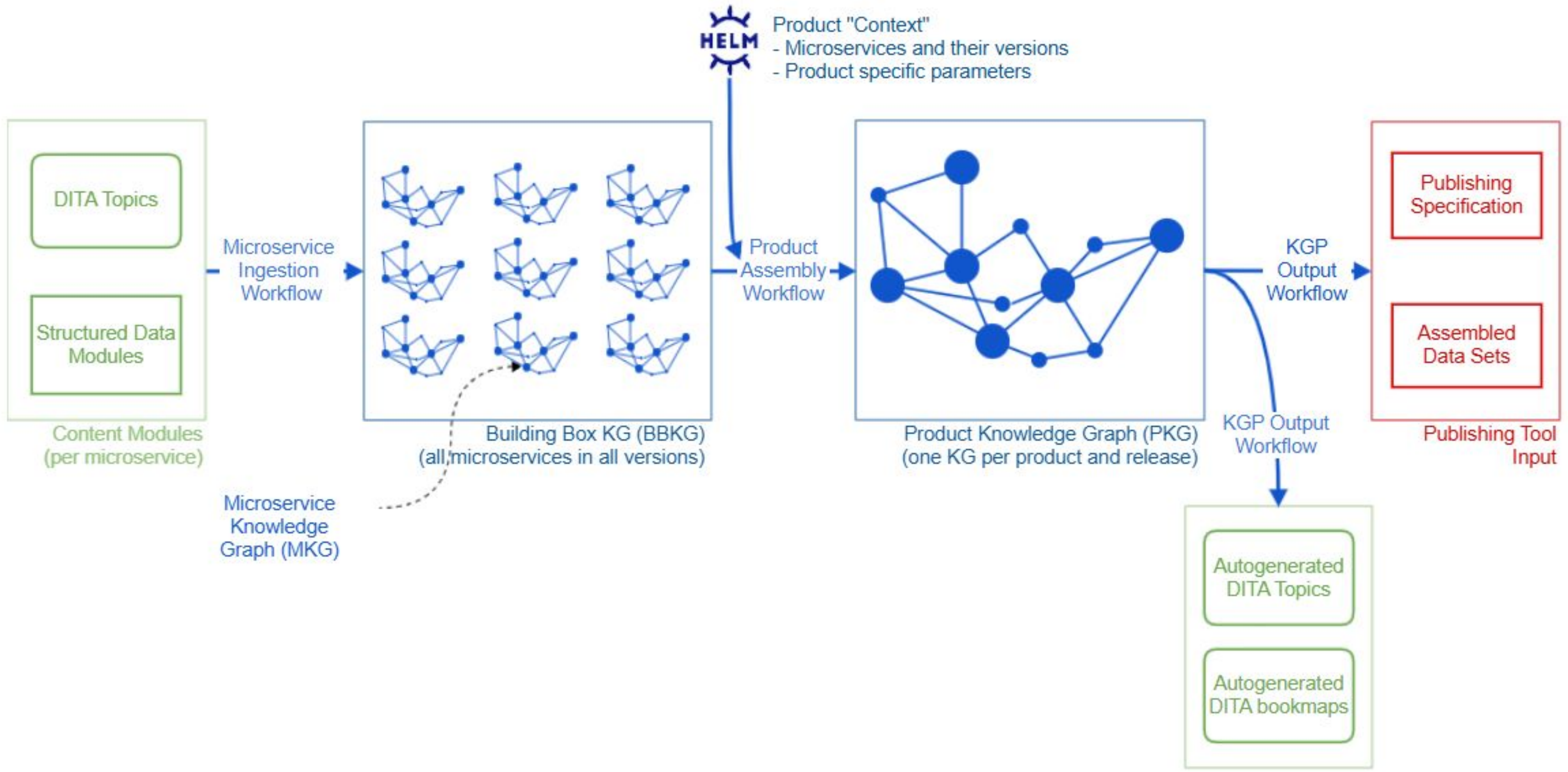
Ontology development:

- Competency questions → graphical model → RDF
- Strive for modular ontologies

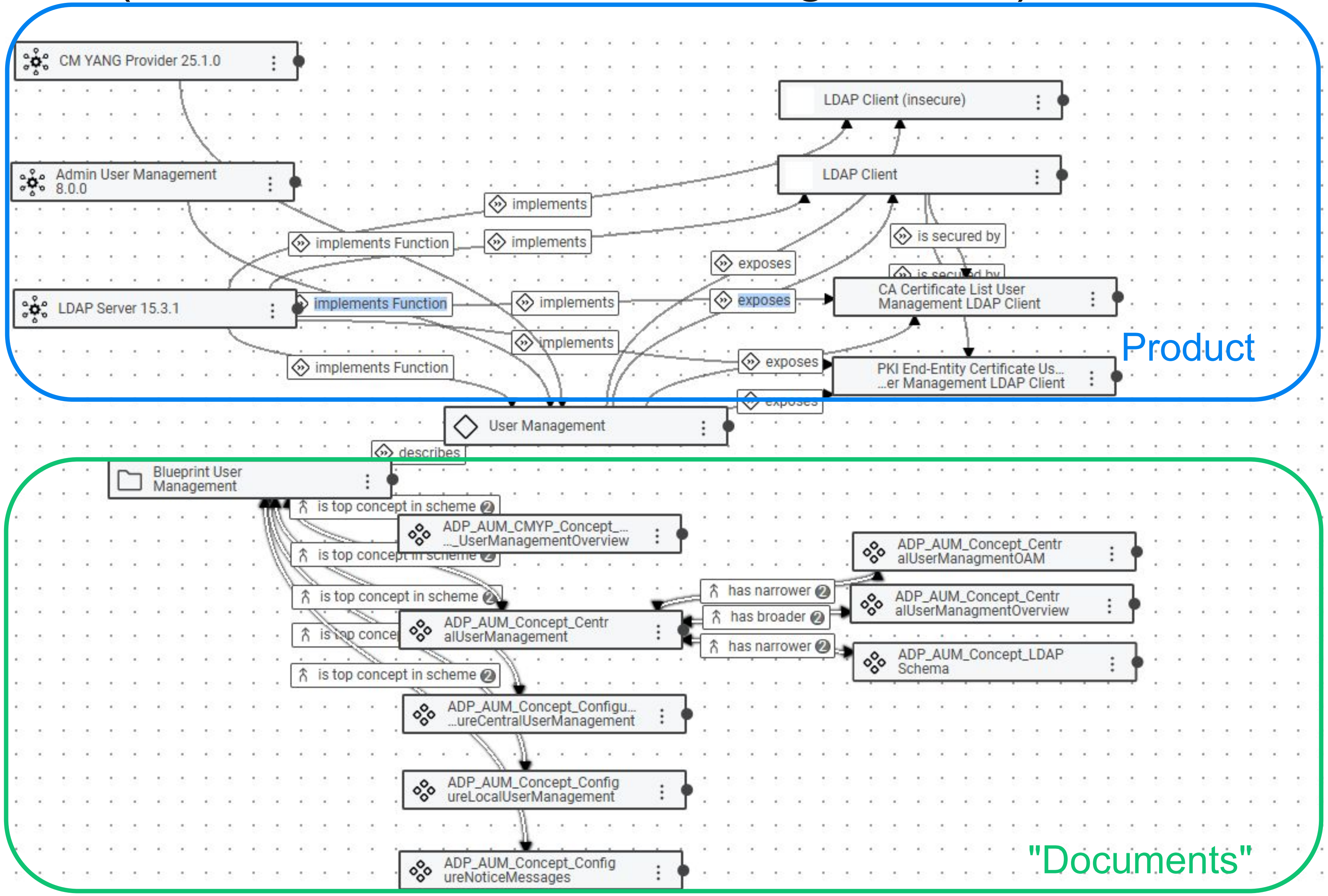


```
# ----------- Interfaces ----------------------------

sys:Interface
    a owl:Class ;
    rdfs:subClassOf sys:System_object ;
    rdfs:isDefinedBy sys: ;
    rdfs:label "Interface"@en .

sys:application_protocol_stack
    a owl:DatatypeProperty ;
    rdfs:isDefinedBy sys: ;
    rdfs:domain sys:Interface ;
    rdfs:range xsd:string ;
    rdfs:label "application protocol stack"@en .

sys:transport_protocol
    a owl:ObjectProperty ;
    rdfs:isDefinedBy sys: ;
    rdfs:domain sys:Interface ;
    rdfs:range sys:Transport_protocol ;
    rdfs:label "transport protocol"@en .

sys:Transport_protocol
    a owl:Class ;
    rdfs:isDefinedBy sys: ;
    rdfs:label "Transport Protocol"@en .
```

Content Delivery Information Flow


HELM
Product "Context"
- Microservices and their versions
- Product specific parameters
DITA Topics
Structured Data Modules
Content Modules (per microservice)
Microservice Ingestion Workflow
Building Box KG (BBKG) (all microservices in all versions)
Microservice Knowledge Graph (MKG)
Product Assembly Workflow
Product Knowledge Graph (PKG) (one KG per product and release)
KGP Output Workflow
Publishing Specification
Assembled Data Sets
Publishing Tool Input
KGP Output Workflow
Autogenerated DITA Topics
Autogenerated DITA bookmaps

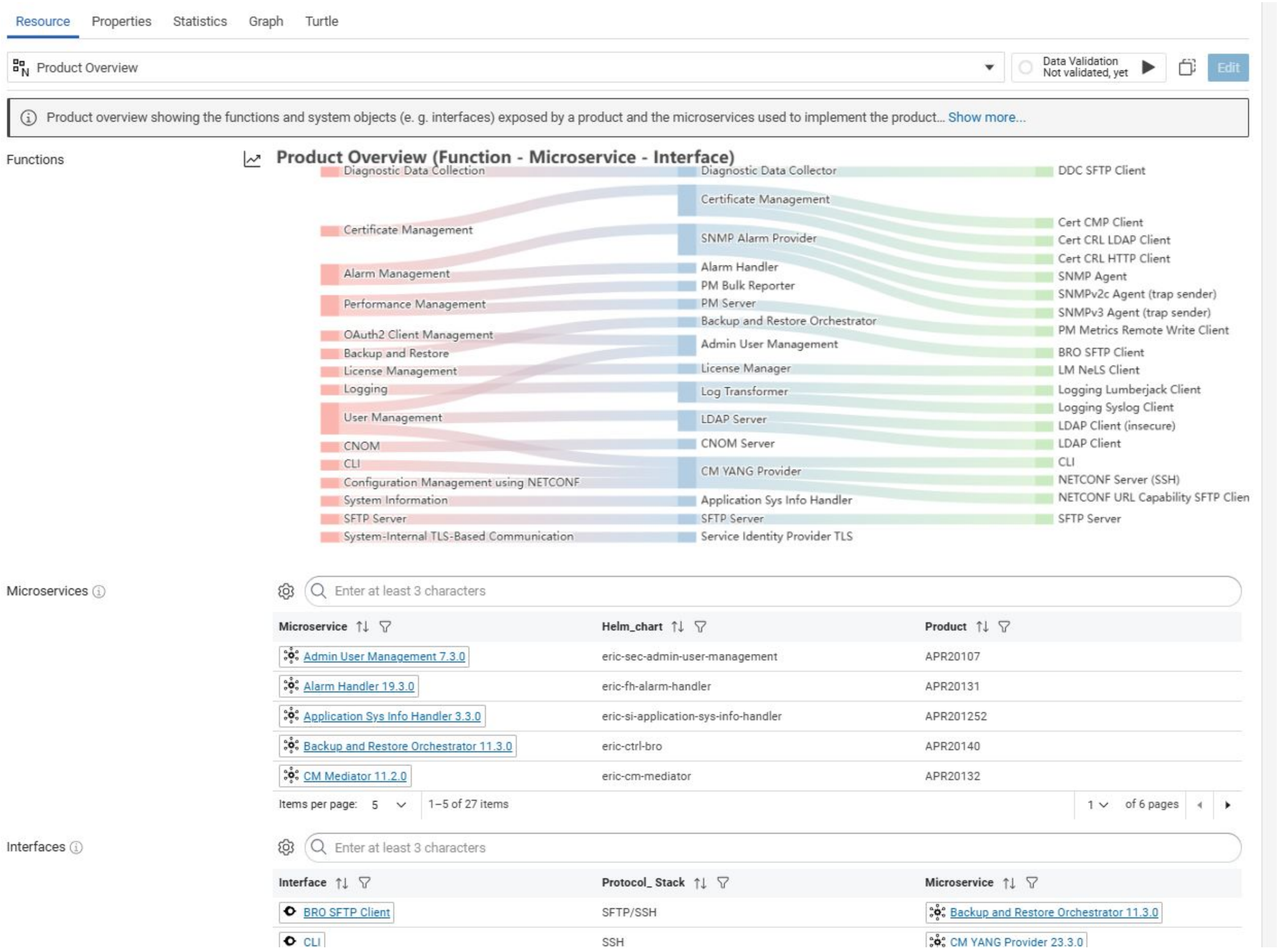
Example: Product Knowledge Graph (PKG)
(eccenca Business Knowledge Editor)



CM YANG Provider 25.1.0
Admin User Management 8.0.0
LDAP Server 15.3.1
LDAP Client (insecure)
LDAP Client
CA Certificate List User Management LDAP Client
PKI End-Entity Certificate Us... ...er Management LDAP Client
implements
implements Function
exposes
is secured by
Product
User Management
describes
Blueprint User Management
is top concept in scheme
ADP_AUM_CMYP_Concept_... ..._UserManagementOverview
ADP_AUM_Concept_CentralUserManagment
ADP_AUM_Concept_CentralUserManagmentOAM
ADP_AUM_Concept_CentralUserManagmentOverview
ADP_AUM_Concept_LDAPSchema
has narrower
has broader
ADP_AUM_Concept_Configu... ...ureCentralUserManagement
ADP_AUM_Concept_ConfigureLocalUserManagement
ADP_AUM_Concept_ConfigureNoticeMessages
"Documents"

- One PKG per product and release
- Small subset of actual PKG shown
- 10s of documents per product (currently)

Example: PKG for System Engineers



Resource | Properties | Statistics | Graph | Turtle

Product Overview — Data Validation: Not validated, yet — Edit

ⓘ Product overview showing the functions and system objects (e. g. interfaces) exposed by a product and the microservices used to implement the product... Show more...

Functions

Product Overview (Function - Microservice - Interface)

Microservices ⓘ

Enter at least 3 characters

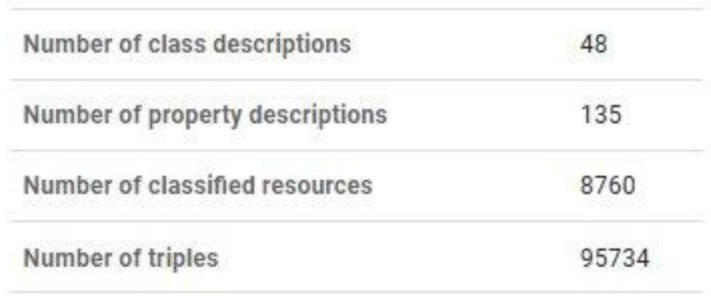
Microservice	Helm_chart	Product
Admin User Management 7.3.0	eric-sec-admin-user-management	APR20107
Alarm Handler 19.3.0	eric-fh-alarm-handler	APR20131
Application Sys Info Handler 3.3.0	eric-si-application-sys-info-handler	APR201252
Backup and Restore Orchestrator 11.3.0	eric-ctrl-bro	APR20140
CM Mediator 11.2.0	eric-cm-mediator	APR20132

Items per page: 5 | 1–5 of 27 items | 1 of 6 pages

Interfaces ⓘ

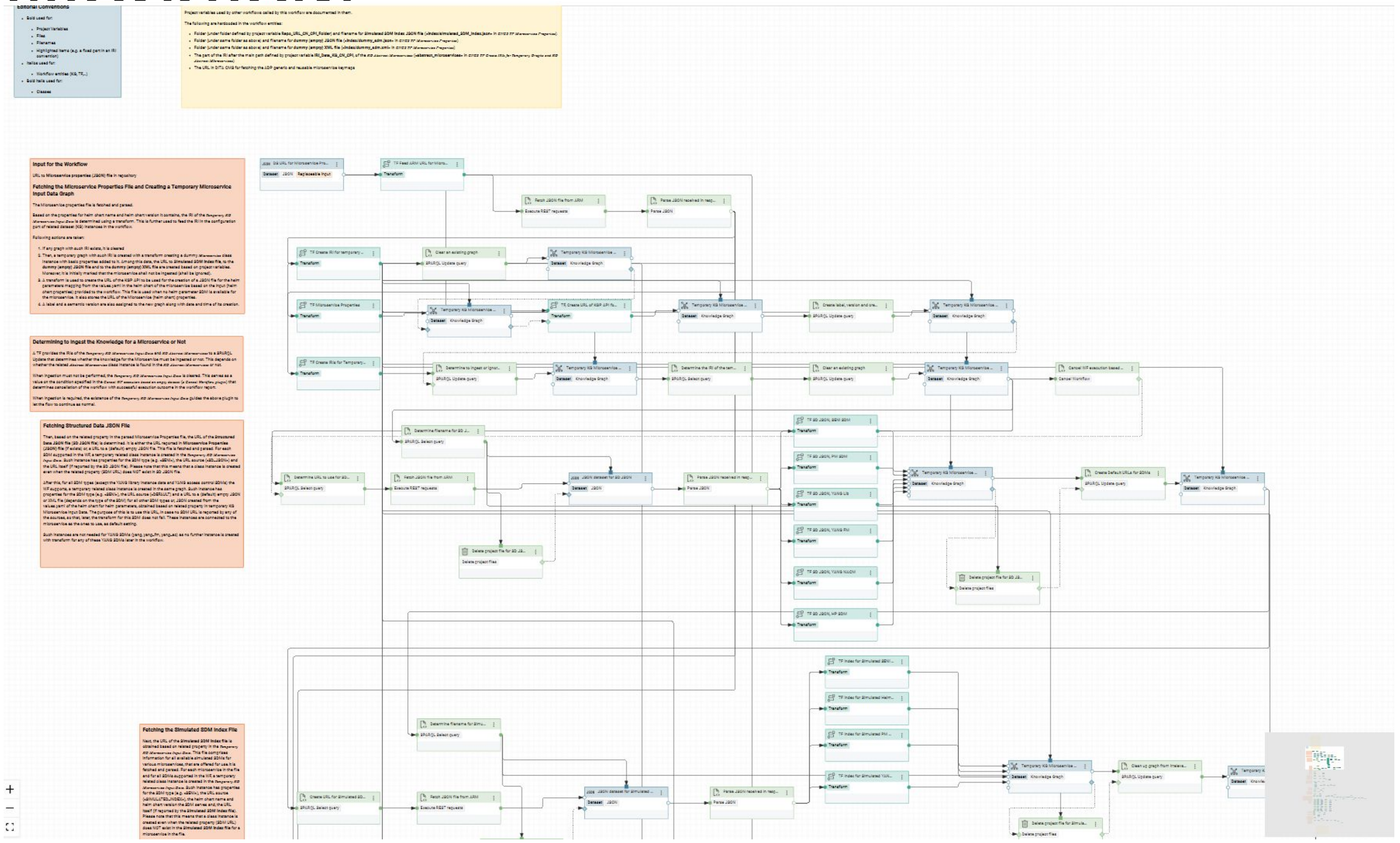
Enter at least 3 characters

Interface	Protocol_ Stack	Microservice
BRO SFTP Client	SFTP/SSH	Backup and Restore Orchestrator 11.3.0
CLI	SSH	CM YANG Provider 23.3.0



Number of class descriptions	48
Number of property descriptions	135
Number of classified resources	8760
Number of triples	95734

Example: eccenca Corporate Memory Workflow



- Screenshot = 10% of Microservice Ingestion WF
- Microservice Ingestion WF: Triggered ~10 times/day, will increase
- Product Assembly and Output WFs: Triggered 2 times/day, will increase
- Created 5 own Python plugins to complement existing plugins

Benefits KG-Based Approach for Tech Docs

Product Manager	Enabler for new product variants
Tech Writer	Less DITA topics, automatic assembly of publishing input
SMEs, Developers	Less support for tech writers
Product Support	Less support questions
Product User	More complete and consistent documentation
GenAI	More complete and consistent documentation
Ericsson	Knowledge Graphs as new data source

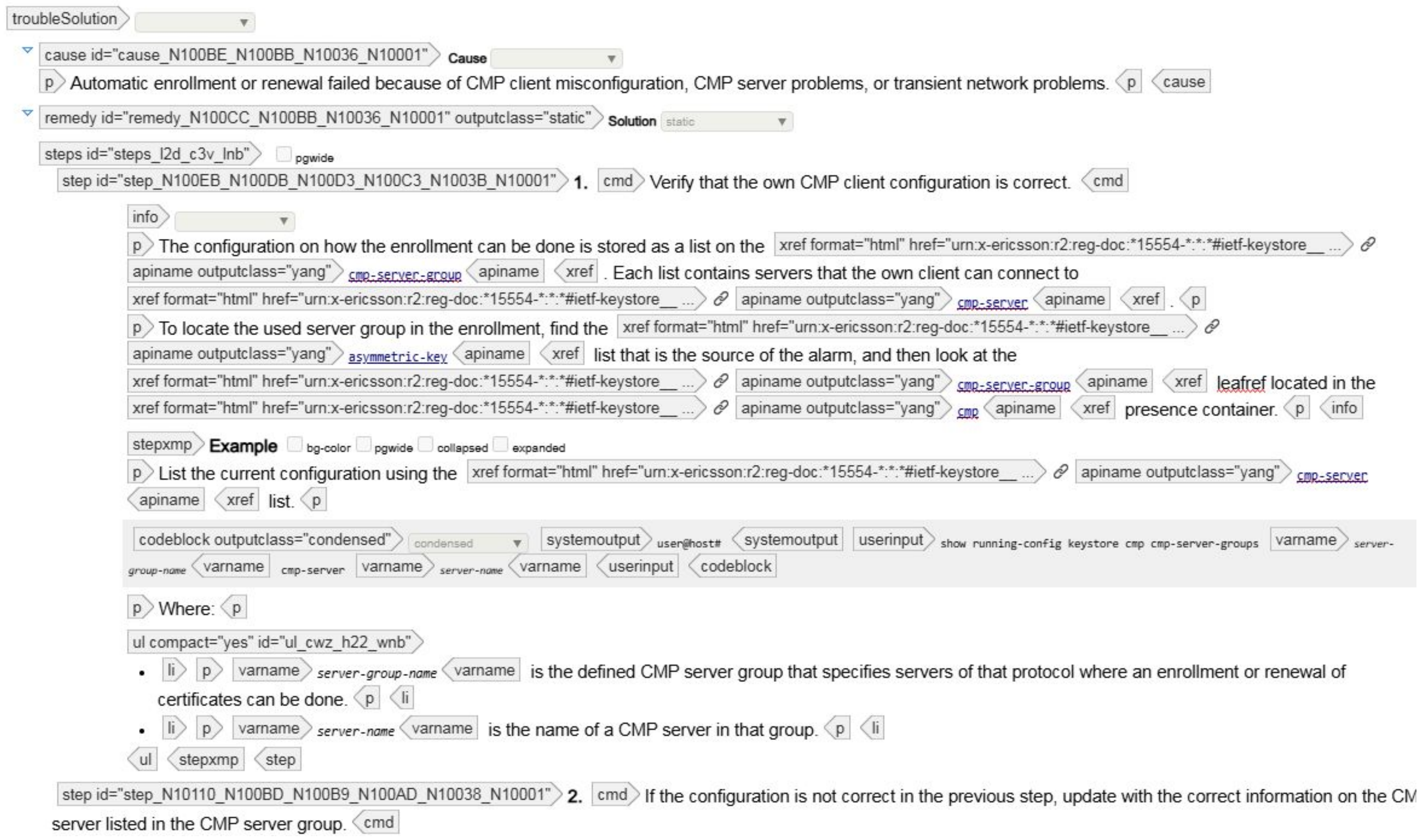
Learnings

- It’s a marathon not a sprint
- Reuse existing artifacts wherever possible
- Only model what you need, not what the involved people know
- Beside good tools you’ll need good people and good data

Questions?

Comments?

DITA Topics: Below the Hood



```
troubleSolution>
  cause id="cause_N100BE_N100BB_N10036_N10001"> Cause
  p> Automatic enrollment or renewal failed because of CMP client misconfiguration, CMP server problems, or transient network problems. <p <cause
  remedy id="remedy_N100CC_N100BB_N10036_N10001" outputclass="static"> Solution static
  steps id="steps_l2d_c3v_lnb"> pgwide
   step id="step_N100EB_N100DB_N100D3_N100C3_N1003B_N10001"> 1. cmd> Verify that the own CMP client configuration is correct. <cmd
        info>
        p> The configuration on how the enrollment can be done is stored as a list on the xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ...>
        apiname outputclass="yang"> cmp-server-group <apiname <xref . Each list contains servers that the own client can connect to
        xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ...> apiname outputclass="yang"> cmp-server <apiname <xref . <p
        p> To locate the used server group in the enrollment, find the xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ...>
        apiname outputclass="yang"> asymmetric-key <apiname <xref list that is the source of the alarm, and then look at the
        xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ...> apiname outputclass="yang"> cmp-server-group <apiname <xref leafref located in the
        xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ...> apiname outputclass="yang"> cmp <apiname <xref presence container. <p <info
        stepxmp> Example bg-color pgwide collapsed expanded
        p> List the current configuration using the xref format="html" href="urn:x-ericsson:r2:reg-doc:*15554-*:*:*#ietf-keystore__ ...> apiname outputclass="yang"> cmp-server
        <apiname <xref list. <p
        codeblock outputclass="condensed"> condensed systemoutput> user@host# <systemoutput userinput> show running-config keystore cmp cmp-server-groups varname> server-
        group-name <varname cmp-server varname> server-name <varname <userinput <codeblock
        p> Where: <p
        ul compact="yes" id="ul_cwz_h22_wnb">
        • li> p> varname> server-group-name <varname is the defined CMP server group that specifies servers of that protocol where an enrollment or renewal of
          certificates can be done. <p <li
        • li> p> varname> server-name <varname is the name of a CMP server in that group. <p <li
        <ul <stepxmp <step
   step id="step_N10110_N100BD_N100B9_N100AD_N10038_N10001"> 2. cmd> If the configuration is not correct in the previous step, update with the correct information on the CM
  server listed in the CMP server group. <cmd
```