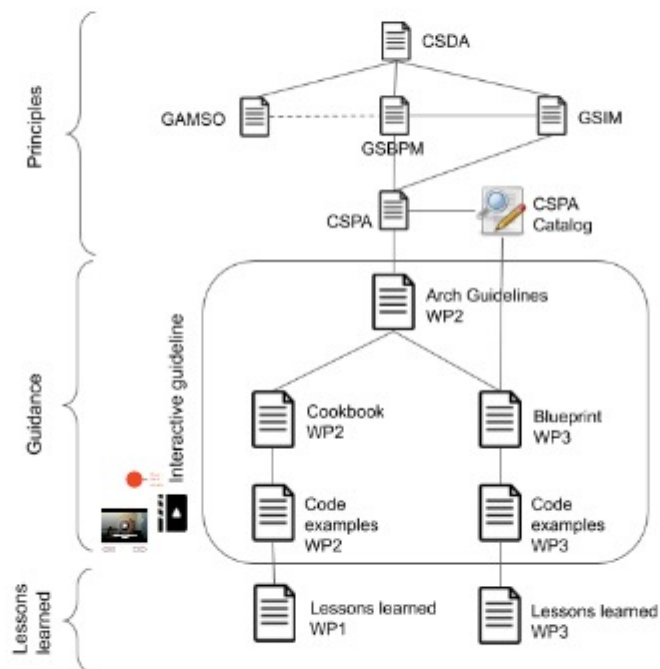


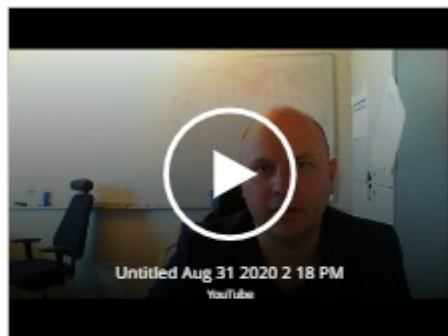
Overview

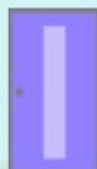




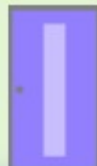
YOU
ARE
HERE

Inter





Production design



Data Collection

CPI

GDP



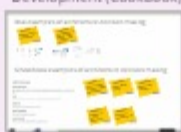
Concepts - "What"



Architecture design - "How"



Development (Cookbook)



Infrastructure design



BUSINESS DRIVERS - "WHY?"

STRONG FINANCIAL PERFORMANCE IS A RESULT OF POSITIVE BUSINESS DRIVERS



STRONG BUSINESS DRIVERS



STRONG BUSINESS DRIVERS



STRONG FINANCIAL PERFORMANCE

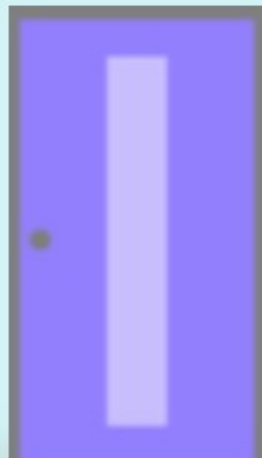
- Revenue
- Profit
- Cash Flow



STRONG BUSINESS DRIVERS



STRONG BUSINESS DRIVERS



BUSINESS DRIVERS - "WHY?"

FROM SUBJECT MATTER ORIENTED TO
SHARED PROCESS ORIENTED SOFTWARE



ENTERPRISE DATA MANAGEMENT

- METADATA
- DATA LAKE
- NEW DATA SOURCES
- HARMONIZING DATA

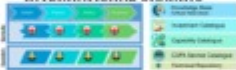


REUSE AND SHARING SERVICES

INTERNALLY SHARING



INTERNATIONAL SHARING



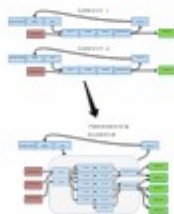
FROM LEGACY TECHNOLOGY
TO SOA AND/OR CLOUD



AGILE BUSINESS
PROCESSES

ADAPT QUICKER TO NEW
POSSIBILITIES AND
THREATS

- NEW DATA SOURCES
- NEW "PRODUCTS" BASED
ON EXISTING DATA
- HARMONIZING STATISTICS
- QUICKLY RESPOND TO NEW
REQUESTS



SPECIFIC ADVANCEMENTS



FROM SUBJECT MATTER ORIENTED TO SHARED PROCESS ORIENTED SOFTWARE



REUSE AND SHARING SERVICES

INTERNALLY SHARING



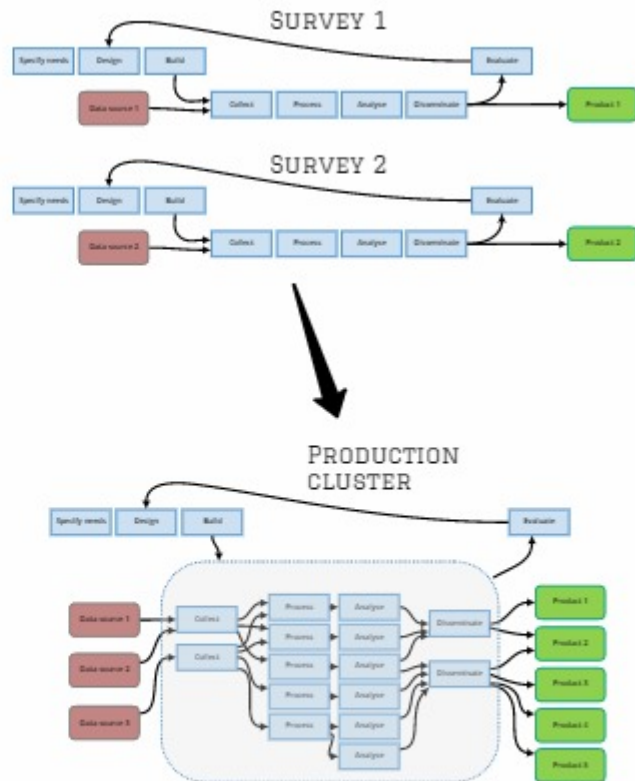
INTERNATIONAL SHARING



AGILE BUSINESS PROCESSES

ADAPT QUICKER TO NEW
POSSIBILITIES AND
THREATS

- NEW DATA SOURCES
- NEW "PRODUCTS" BASED
ON EXISTING DATA
- HARMONIZING STATISTICS
- QUICKLY RESPOND TO NEW
REQUESTS

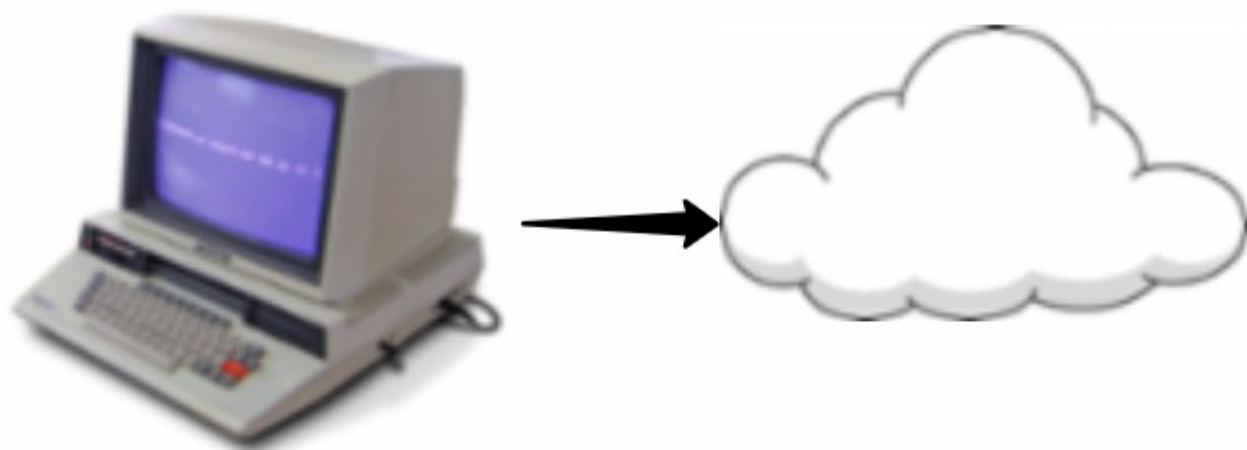


ENTERPRISE DATA MANAGEMENT

- METADATA
- DATA LAKE
- NEW DATA SOURCES
- HARMONIZING DATA



FROM LEGACY TECHNOLOGY TO SOA AND/OR CLOUD



SPECIFIC ADVANCEMENTS



SECURITY



USER EXPERIENCE



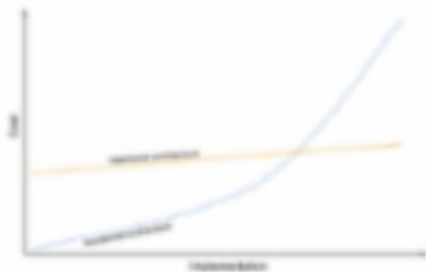
INTEGRATION



OPEN DATA

STRATEGY

ARCHITECTURE



APPLICATION PORTFOLIO



CHANGE MANAGEMENT

- BUSINESS AREA
- STAFF / ROLES
- DEPARTMENT
- LOW-HANGING FRUIT
- SPECIFIC KPI

INVESTMENT POSSIBILITY

80 / 20 Rule



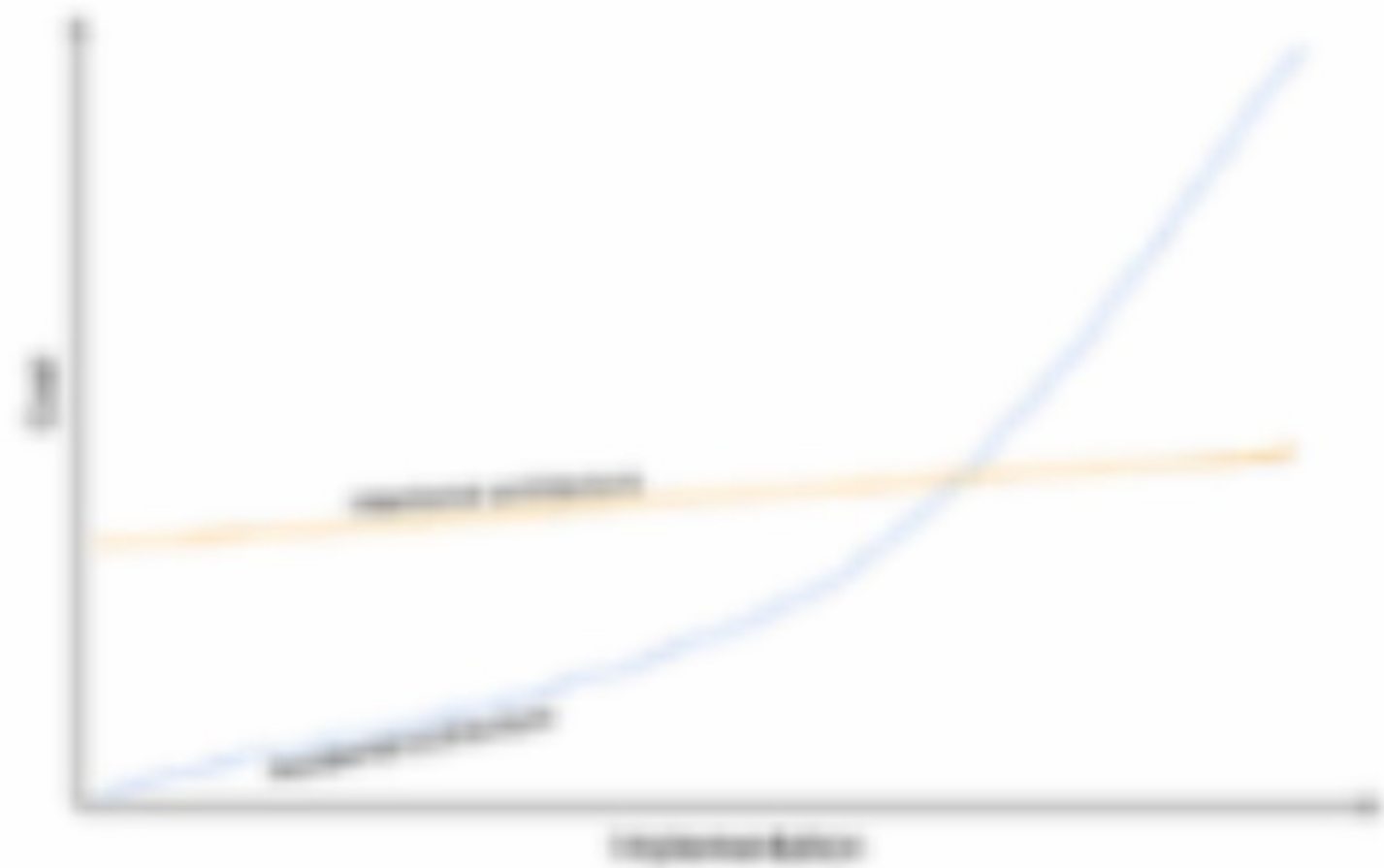
RISK APPETITE

TRENDS



- DATA SCIENCE
- BIG DATA
- AI / ML
- LOW CODE
- HYBRID CLOUD

ARCHITECTURE



APPLICATION PORTFOLIO

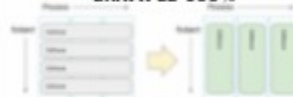
STABILITY



INVESTMENT POSSIBILITY

80 / 20 RULE

EXAMPLE 100%



EXAMPLE 80%



CHANGE MANAGEMENT

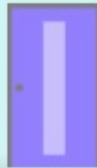
- BUSINESS AREA
- STAFF / ROLES
- DEPARTMENT
- LOW-HANGING FRUIT
- SPECIFIC KPI

RISK APPETITE

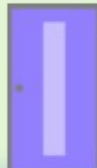
TRENDS



- DATA SCIENCE
- BIG DATA
- AI / ML
- LOW CODE
- HYBRID CLOUD



Production design



Data Collection

CPI

GDP



Concepts - "What"



Architecture design - "How?"



Development (Cookbook)



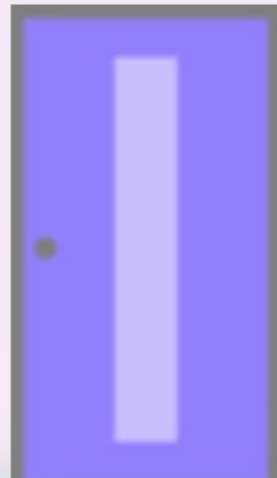
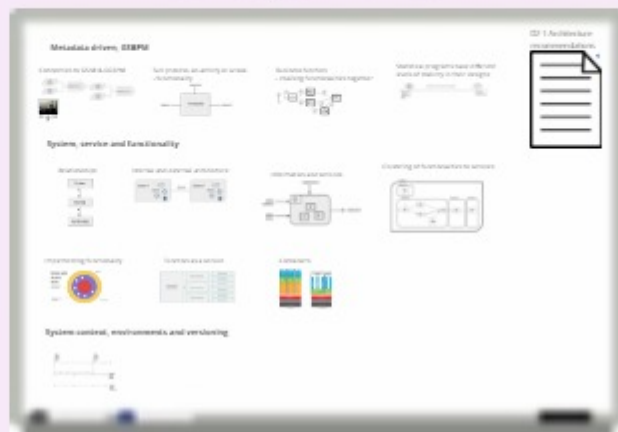
Infrastructure design



Reference models



Concepts - "What"



Reference models



Concepts - "What"

Metadata driven, GSBPM

Connection to GSIM & GSBPM



Sub process, an activity or a task
- functionality



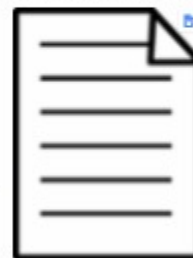
Business function
- chaining functionalities together



Statistical programs have different levels of stability in their designs



D2-1 Architecture recommendations



System, service and functionality

Relationships



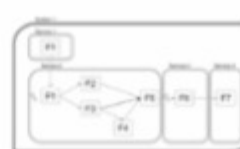
Internal and external architecture



Information and services



Clustering of functionalities to services



Implementing functionality



Function as a service



Containers

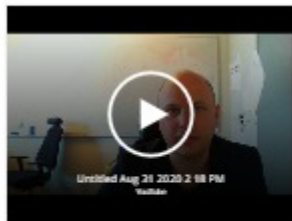


System context, environments and versioning



Metadata driven, GSBPM

Connection to GSIM & GSBPM



Sub process, an activity or a task
- functionality



Business function

- chaining functionalities together



Statistical programs have different levels of stability in their designs



System, service and functionality

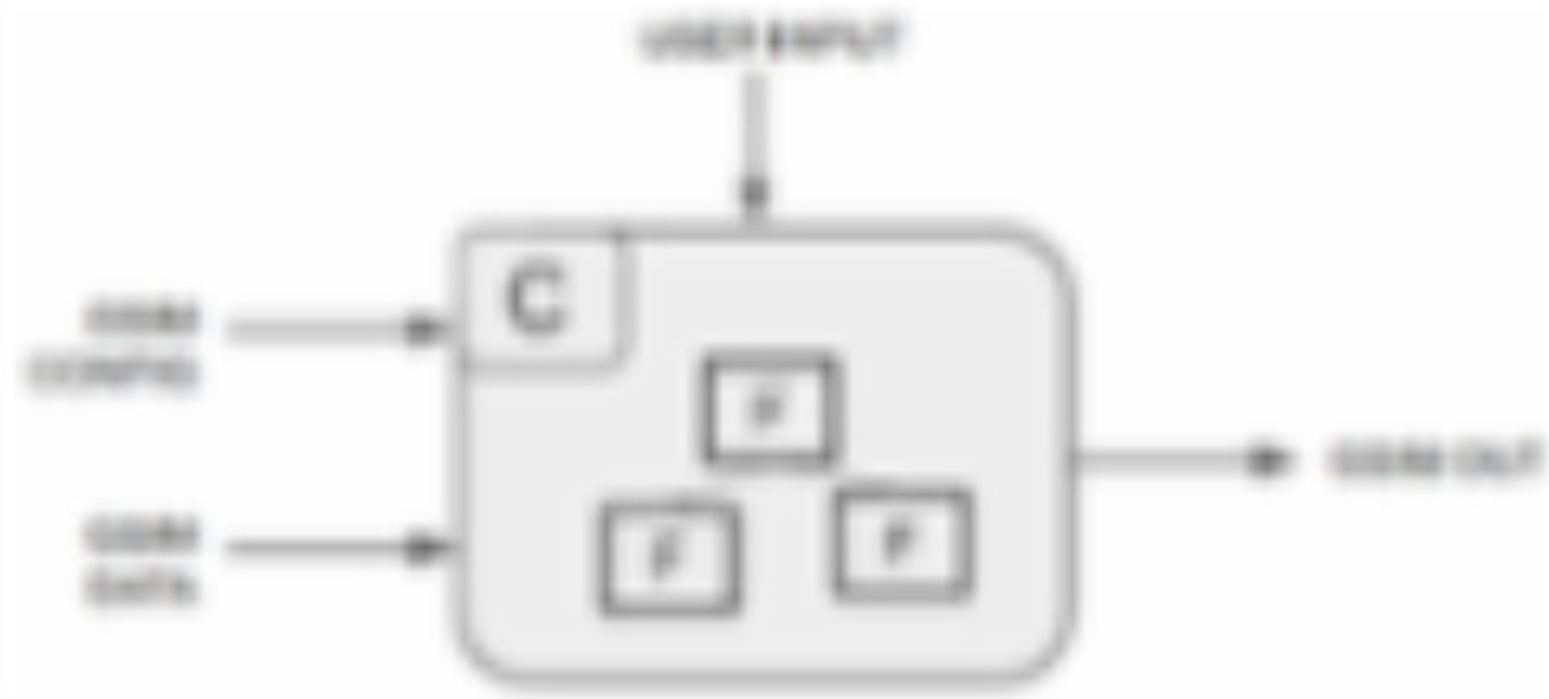
Relationships



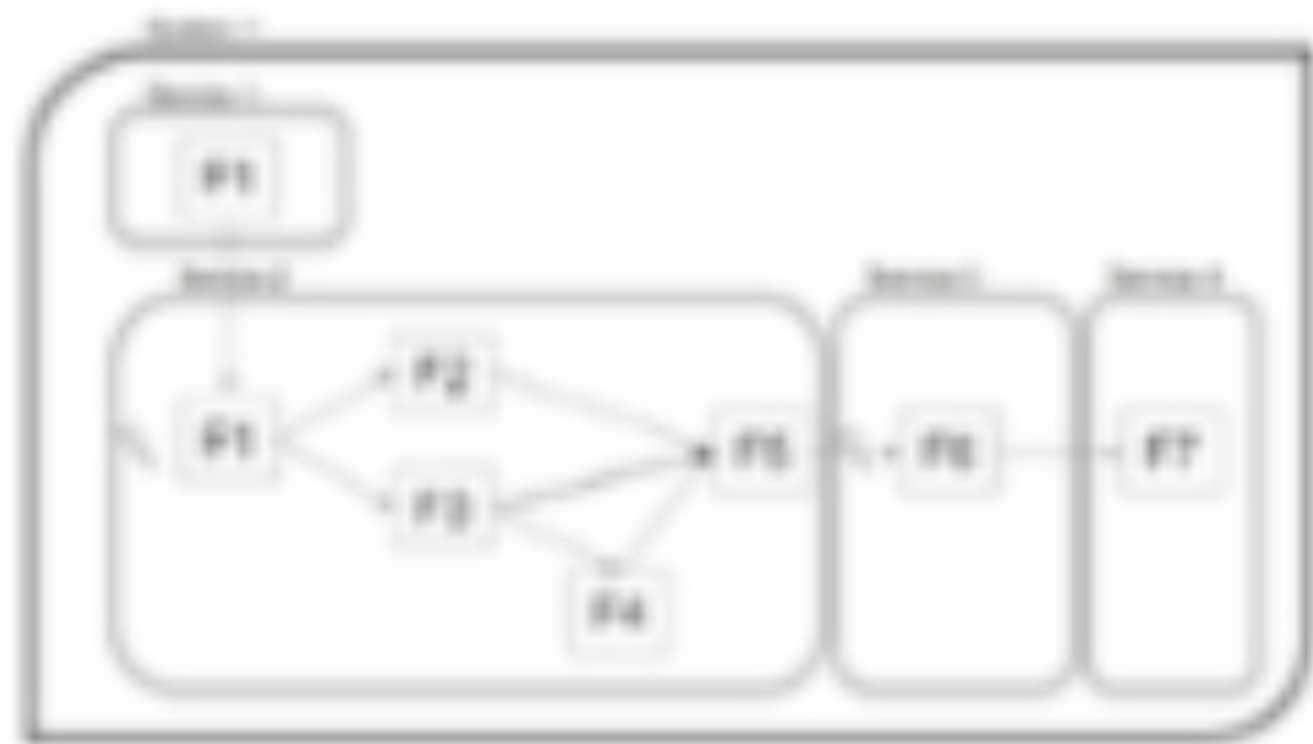
Internal and external architecture



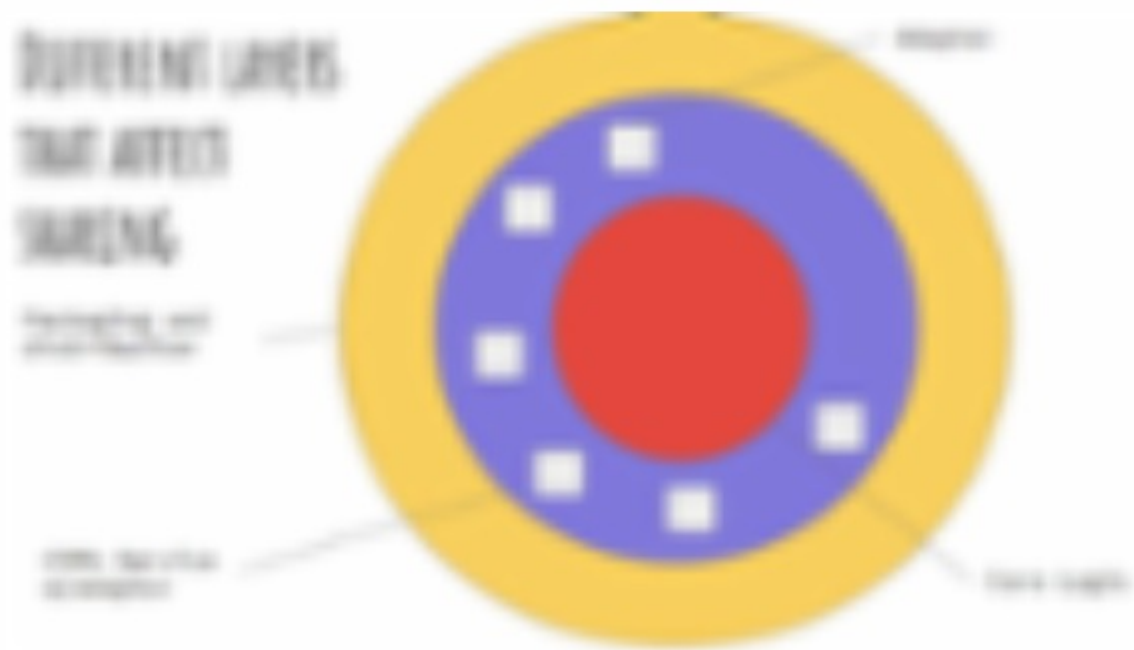
Information and services



Clustering of functionalities to services



Implementing functionality



Function as a service



Containers



System context, environments and versioning

1000
100
10
1



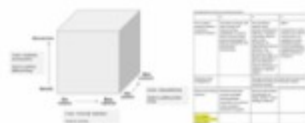
Architecture design - "How?"

Sandboxing for exploration

Security



Performance and Scalability



Resilient services and error handling

Versioning

- Service versioning
- Endpoint versioning
- GDM structure versioning
- Information object instance versioning



Deployment

Containerization

Design principles for service autonomy

Data management

Integration patterns



Multilingual support

Open source

Moving from legacy architecture to service oriented architecture

System/Application/Data

Containerisation

Host OS

Virtual Infrastructure

Physical Infrastructure -
Server/Storage/Network

System/Application/Data

Containerisation

Host OS

Virtual Infrastructure

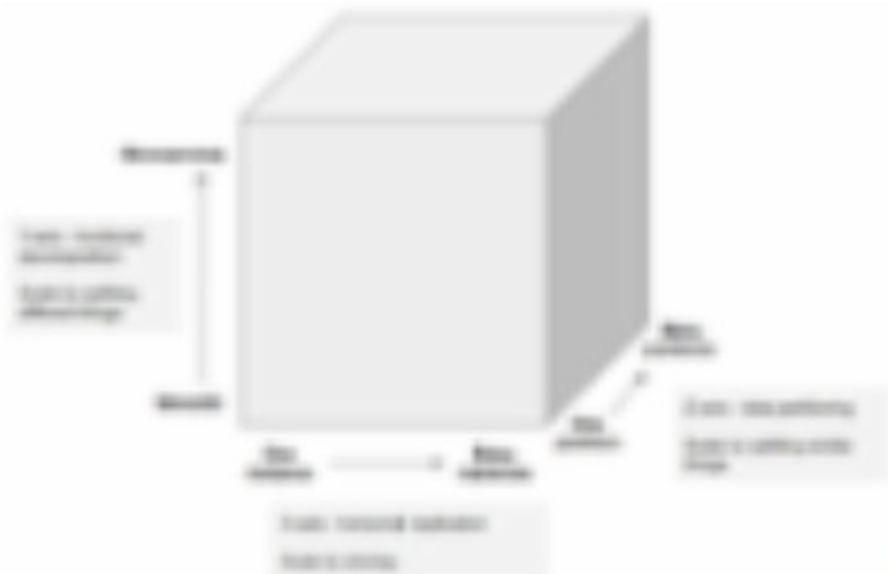
Physical Infrastructure -
Server/Storage/Network

Sandboxing for exploration

Security



Performance and Scalability



Considerations along the business domain

	1	2	3
From subject customer software to provide customer software	Increased customer and data volume will require scaling capabilities. It will be with all users scaling with the performance in terms of availability and performance. Availability	Increased customer and data volume will require scaling the software. Provide supporting software with all users performance will be supported by software service. The availability of the service will differ from other supporting software. The service will be a single unit.	Step 2 Increase the customer volume in the year the scaling takes. For example, the data availability will be supported in agreement with the software hardware Programs.
Enterprise data management		Storage solutions with high data volume and increased number of customers needs enterprise scaling	
Mobile and cloud services	Workload and data volume will differ significantly between all services. Scalability is dependent on support	Service will require scalability and data storage. Mobile scaling is critical	
High Availability High Scalability High Performance			

Resilient services and error handling

Versioning

- Service versioning
- Endpoint versioning
- GSIM structure versioning
- Information object instance versioning



Deployment

Containerization

Design principles for service autonomy

Data management

Integration patterns



	Business goals strategies	Business information	"Smart Services"
People	<ul style="list-style-type: none"> • Increase sales 	<ul style="list-style-type: none"> • Customer segment • Location • Demographics 	<ul style="list-style-type: none"> • Cross-selling • Up-selling • Personalized recommendations • Chat
Business	<ul style="list-style-type: none"> • Increase market share, revenue, loyalty • Increase customer engagement and retention 	<ul style="list-style-type: none"> • High level summary • Location • Time of day and day of week • Weather and other contextual data 	<ul style="list-style-type: none"> • Hyper-personalized content • Location-based targeting and offers • Behavioral and contextual targeting
Engagement actions	<ul style="list-style-type: none"> • Increase sales volume • Increase the size of the basket 	<ul style="list-style-type: none"> • Increase the size of the basket • Increase the size of the basket 	<ul style="list-style-type: none"> • Hyper-personalized content • Location-based targeting and offers

Multilingual support

Open source

Moving from legacy architecture to service oriented architecture

Development (Cookbook)

Real examples of architecture decision making

EXAMPLE 1

IMPLEMENTATION OF
CONTAINERIZATION



EXAMPLE 2

IMPLEMENTATION OF
METADATA DRIVEN
VALIDATIONS WITH
ADAPTORS



Schoolbook examples of architecture decision making

FLOR - the service

Service of Program

This service is the main service in the system, responsible for the main logic of the system.

Class

This service is the main service in the system, responsible for the main logic of the system.

Class use

Service for managing and setting up the system.

Error Localization

Service for error localization in the system.

Manual setting

Service for manual setting in the system.

SCENARIO 1

FROM DISCONNECTED
SERVICES TO CONNECTED
METADATA DRIVEN

SCENARIO 2

CONTEXT AWARE
SERVICES

SCENARIO 3

REVISITING DATA
INTEGRATION

SCENARIO 4

CONTAINERS AND
MULTIPLE
ENVIRONMENTS

SCENARIO 5

DESIGN-DRIVEN
INFORMATION FLOWS

Real examples of architecture decision making

EXAMPLE 1

IMPLEMENTATION OF CONTAINERIZATION



EXAMPLE 2

IMPLEMENTATION OF METADATA DRIVEN VALIDATIONS WITH ADAPTORS



EXAMPLE 1

IMPLEMENTATION OF CONTAINERIZATION



EXAMPLE 2

IMPLEMENTATION OF METADATA DRIVEN VALIDATIONS WITH ADAPTORS



Schoolbook examples of architecture decision making

Flight - the service

Service Program

The program of the service is a set of rules that governs the behavior of the service.

Code

The code is the implementation of the service.

Configuration

The configuration is the set of parameters that control the service.

Error Handling

The error handling is the set of rules that governs the behavior of the service in the event of an error.

Manual Testing

The manual testing is the set of rules that governs the behavior of the service in the event of a manual test.

SCENARIO 1

FROM DISCONNECTED
SERVICES TO CONNECTED
METADATA DRIVEN

SCENARIO 2

CONTEXT AWARE
SERVICES

SCENARIO 3

REVISITING DATA
INTEGRATION

SCENARIO 4

CONTAINERS AND
MULTIPLE
ENVIRONMENTS

SCENARIO 5

DESIGN-DRIVEN
INFORMATION FLOWS

FLIGHT - the services

Initial program

After departure of the aircraft, the cabin crew will be responsible for ensuring the cabin is secure and the aircraft is ready for the next flight.

1. Cabin

The cabin is the area of the aircraft where the passengers are seated and the cabin crew are responsible for ensuring the cabin is secure and the aircraft is ready for the next flight.

2. Cabin crew

The cabin crew are responsible for ensuring the cabin is secure and the aircraft is ready for the next flight.

3. Cabin crew

The cabin crew are responsible for ensuring the cabin is secure and the aircraft is ready for the next flight.

4. Cabin crew

The cabin crew are responsible for ensuring the cabin is secure and the aircraft is ready for the next flight.

SCENARIO 1

FROM DISCONNECTED
SERVICES TO CONNECTED
METADATA DRIVEN

SCENARIO 2

CONTEXT AWARE
SERVICES

SCENARIO 3

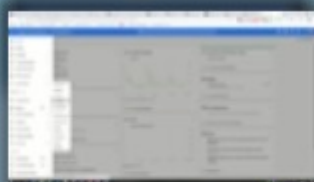
REVISITING DATA
INTEGRATION

SCENARIO 4

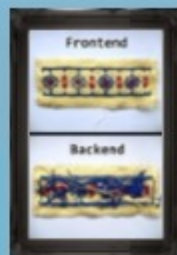
CONTAINERS AND
MULTIPLE
ENVIRONMENTS

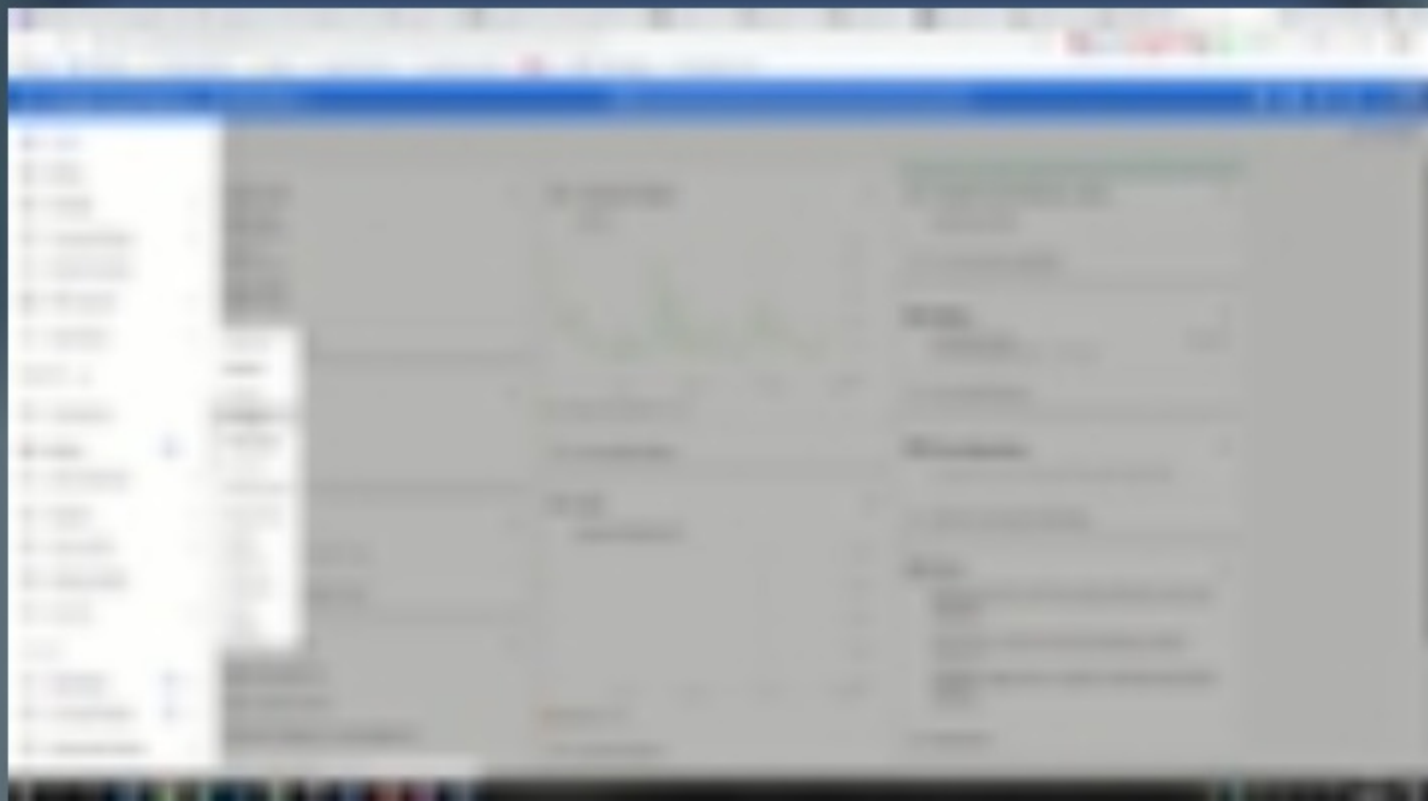
SCENARIO 5

DESIGN-DRIVEN
INFORMATION FLOWS



Frontend
Backend
Database
API
UI
UX
Design
Development
Testing
Deployment
Maintenance
C





SCENARIO 1

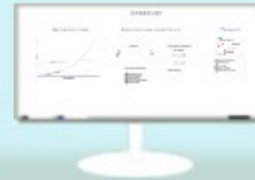
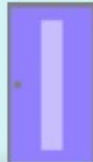
FROM DISCONNECTED
SERVICES TO CONNECTED
METADATA DRIVEN



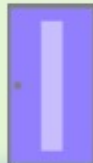
SCENARIO 1

FROM DISCONNECTED
SERVICES TO CONNECTED
METADATA DRIVEN





Production design



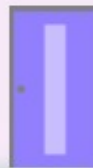
Data Collection

CPI

GDP



Concepts - "What"



Architecture design - "How"



Development (Cookbook)



Infrastructure design



