Event-Driven architectures in practice

Ewout Devos & Jorn Othmer
Introduction

What is event driven integration?
“The event-driven architecture pattern is a popular distributed asynchronous architecture pattern used to produce highly scalable distributed applications”
Request-driven way

App A

App B
App C
App D

I tell others what to do

Event-driven way

App A

App B
App C
App D

Event broker

I broadcast what I did

I work out what to do next

we commit. we deliver.
The Pizza Analogy

1. Tables
2. Kitchen
3. Stock
4. Cashier

we commit. we deliver.
The Pizza Analogy

Tables

Kitchen

Stock

Cashier

Pepperoni for table 6

Notification

Notification

Notification

Event Broker

6 we commit. we deliver.
Advantages of Event-Driven Architectures

Loose Coupling
No logic needed associated with other services

Fault tolerance
If a subscribing service fails, it doesn't impact the producer of the event

Real-time integration
Events are recorded as they occur

Increased versatility
Components function independently and can be used and reused many times over
Disadvantages of Event-Driven Architectures

Need for standardization
Required to get loosely coupled events

Monitoring
Do we have a central location to have an overview?
How will we troubleshoot in case of errors?

Governance
Sometimes sequential processing is required
Use Case

Event driven integration with our customer: Edgard and Cooper.
About Edgard & Cooper

- Belgian company that makes dog & cat food.
- Founded in 2016.
- Over 100 employees by now.
- Available by now in over 12 countries.
- Delivers 20,000 kg of pet food each day.
- Over 45 products
- Sales almost doubled compared to last year.

⇒ Time for a new ERP system!
Event driven O2C enterprise landscape

**Requirements:**
- Nearly real-time
- Scalable
- Robust
- Flexible

**Shopify:**
- B2C Salesorders registered in Shopify
- Registration in S4
- Feedback in Shopify for:
  - Salesorder
Event driven O2C enterprise landscape

Components:
- SAP Event Mesh (EM) Queues & Topics
- SAP Cloud Integration (SCI)
- Shopify API's (Webshop)
- SAP S/4HANA CLOUD

Flow:
1. Post event to Topic.
2. Subscribe receiver’s Queue to Topic.
3. Process & retrieve additional info from source.
4. Deliver message to target.
Event driven O2C enterprise landscape

In detail:

- **Shopify <> S4HC**
  - **Shopify:**
    - Sends SO notification.
    - Makes SO available in API.
    - Retreives SO confirmation from EM.
  - **S4:**
    - Receives SO from CPI.
    - Sends Succes/Fail for SO.
Event driven O2C enterprise landscape

» Salesorder Create

Shopify posts event on EM topic

SCI picks up SO event from queue

S4HC receives sales order

SCI retrieves Sales Order from Shopify

we commit. we deliver.
Event driven O2C enterprise landscape

- Sales Order Success/Fail Event

S4HC posts event on EM topic

SCI Picks up SO event from queue

SCI posts notification on EM topic

Shopify reads SO notification queue

S4HC

EM Topic Queue

SCI

EM Topic Queue

Shopify

we commit. we deliver.
Use Case - demo
Sales order Creation & Confirmation
between S/4 Hana Cloud and the Shopify B2C webshop.
Challenges

Event driven integration with our customer: Edgard and Cooper.
Challenges

- AMQP Adapter
- Monitoring S4 – EM
- Dead letter queue
- Standardize events
Conclusion

Event driven integration with our customer:
Edgard and Cooper.
Conclusion

- Event-based architecture to decouple systems
- Built a scalable and dynamic architecture to handle the project requirements
- Some challenges, but overcame together with SAP
Useful links

Event-Driven Integrations Using SAP Cloud Platform Integration Suite | SAP TechEd 2020
• https://www.youtube.com/watch?v=98FNX-HFdGg

SAP Event Mesh
• https://help.sap.com/viewer/bf82e6b26456494cbdd197057c09979f/Cloud/en-US/df532e8735eb4322b00bfce42f84e8d.html

Ewout Devos
• Ewout.Devos@delaware.pro

Jorn Othmer
• Jorn.Othmer@delaware.pro
Questions?

we commit. we deliver.