

DEVINITI



Platinum  
Solution Partner  
ENTERPRISE

---

# Foundations of point-to-point application integrations



Łukasz Krupa

# Agenda

---

## Introduction to an app integration

Definition, Popular models, Why we need it?

## Why integrations are difficult?

7 layeres

## How to do them right?

7 tips

# What is an app integration?

---

When two applications:

- Exchange data (attachments, tickets, comments)
- Triggers some actions (workflow transition)
- Shares configuration (user directory)

# Flavours of integrations

Point-to-point

EAI

SOA/ESB

Microservices

Internet of things & wearables

API-centric

iPaaS

# Why we need integrations?

Automation

Speed, efficiency

Reliability



# World without NEED of app integration







# Why integrations are difficult?

Involves more people/teams

More complex than other app changes

Higher risk during operation phase





**7** layers something can go wrong

# Business Layer

1 / 7

*Will not get full month report*

- > Kill performance
- > API not available
- > Collect & sent data to cover reporting needs



# Process/Workflow Layer

2/7

*My „closed” status means „resolved”.*

*After 24h system closes the ticket if no complaints.*



# Application Layer

3/7

- Transition with screen + CF required or missing on screen
- Authentication failed (password expired)
- Data transformation (convert dates to text)

# Platform Layer (app server)

4/7

- JVM version not supported by integrated (external) system library / driver
- Protocol not supported (SOAP, RPC)

# Compliance Layer

5/7

- SSL certification expires
- GDPR (username)

# OS & Server Layer

6/7

- Firewall
- Antivirus
- System patching

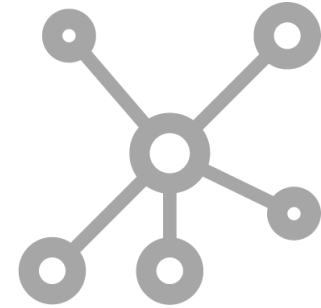




# Network Layer

7/7

- Latency, timeouts
- DNS
- Lost communication, transaction interruption



# Why integrations are difficult?

Comparing to typical app changes there are 3-4 layers instead of 7.



**7** tips

**How to do better app integrations**

# Your system is down

1 / 7

- Not processed messages
  - > Are they stored in memory, files, db?
- State of exchanged data transaction
- Switching to standby server
  - > Can you establish new connection with remote system?
  - > Can you handle remote responses & errors that are part of unfinished primary server operation?

# Remote system is down

2/7

- Can you keep unsent messages forever?
- Can you retry?
  - > Can you send messages in right order?
- Is communication Sync or Async?

# Describe your solution

3/7

- Document architecture
  - > Include key components where messaging flow can be interrupted and monitored (DB, file system server, messaging server).
  - > Use Enterprise Integration Patterns
- Make sure everyone will understand it
  - > Diagrams, UML

# Integration & end-to-end testing

4/7

- Keeps your integration safe when implementing other app changes



# Record any obstacles

5/7

- From development and testing phase
  - > People change often
  - > For some IT teams your integration is maybe a small task on their list
- Your team will change as well
  - > developers will transition to support team

E. g. SSL certification expires on test env. after a while

# Clearly define responsibility

6/7

- Integration is not working? Who should start first..?
- Create ordered checklist
- Create contact list

# Monitoring & alerting

7<sub>17</sub>

- Wide as possible
- Try to alert before problem materialize  
Too many messages in queue, timeout frequency
- Include your partner

# Summary

---

- Shades of application integration
- We will integrate applications more and more
  - More automated and integrated solutions
  - We are NOT „Model-T Sociaty” alike
- Integration is not an easy tasks
- How to do it right



DEVINITI



Platinum  
Solution Partner  
ENTERPRISE

---

Thank you!



Łukasz Krupa