Will trade an ESB for an agile Integration solution in the Cloud

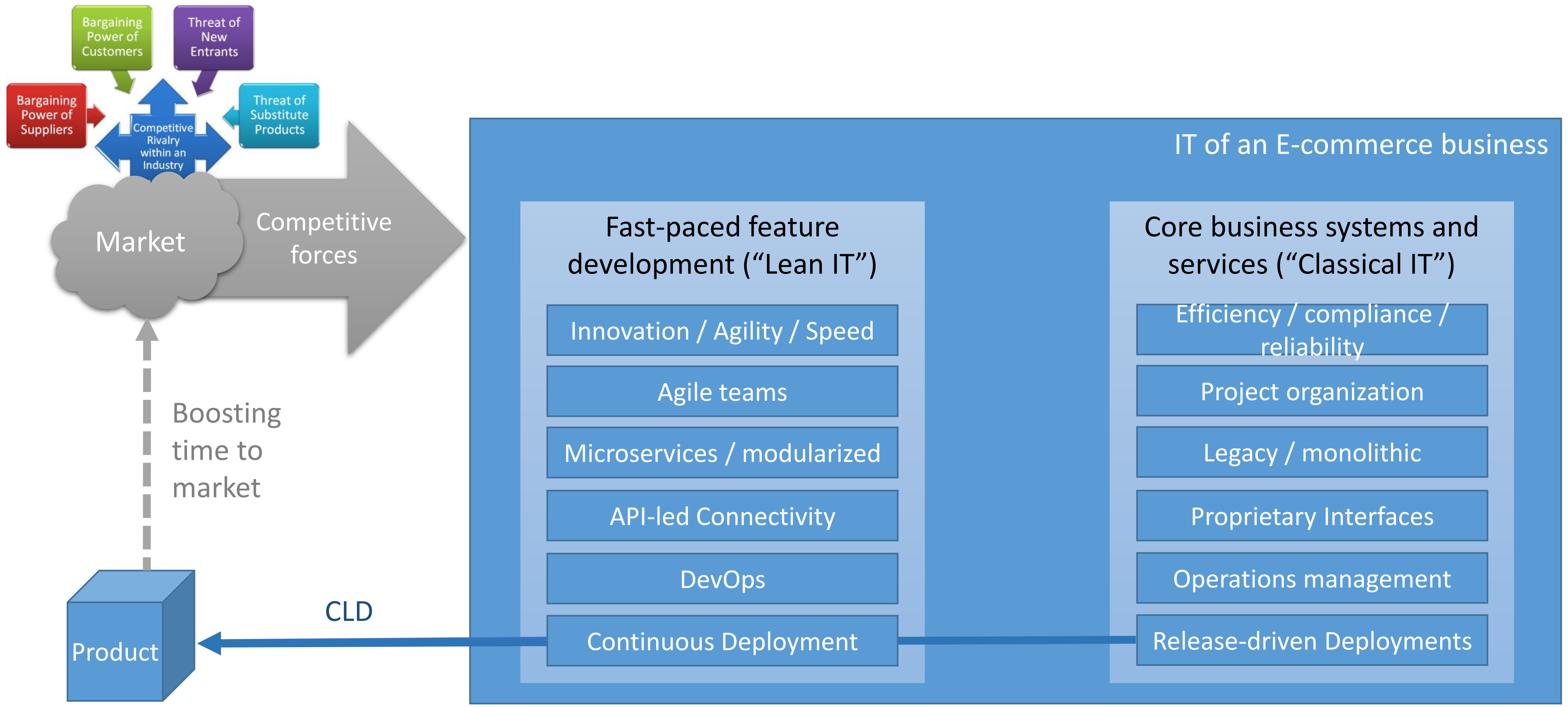
Berlin Expert Days,
16. September 2016, Urania Berlin

@KayLerch | Engineering Manager | Immobilien Scout

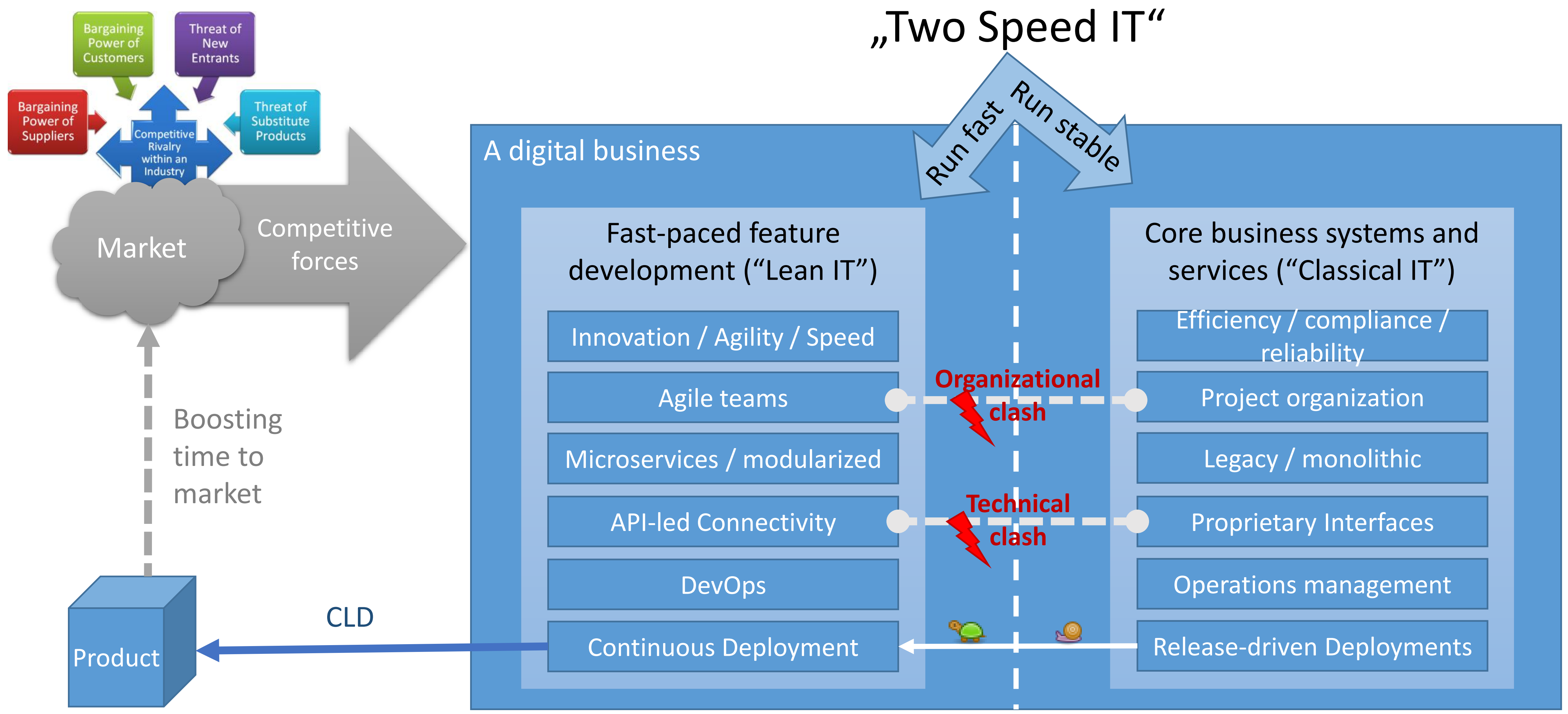
Agenda

- **Disruptive forces and what they do with enterprise IT**
- An ideal integration platform
- AWS Simple Workflows (SWF) in a nutshell
- Demo time
- Leveraging SWF to get rid of a classical ESB solution
- Reclaim process ownership and end-2-end-autonomy
- Drawing the big picture of a hybrid integration solution

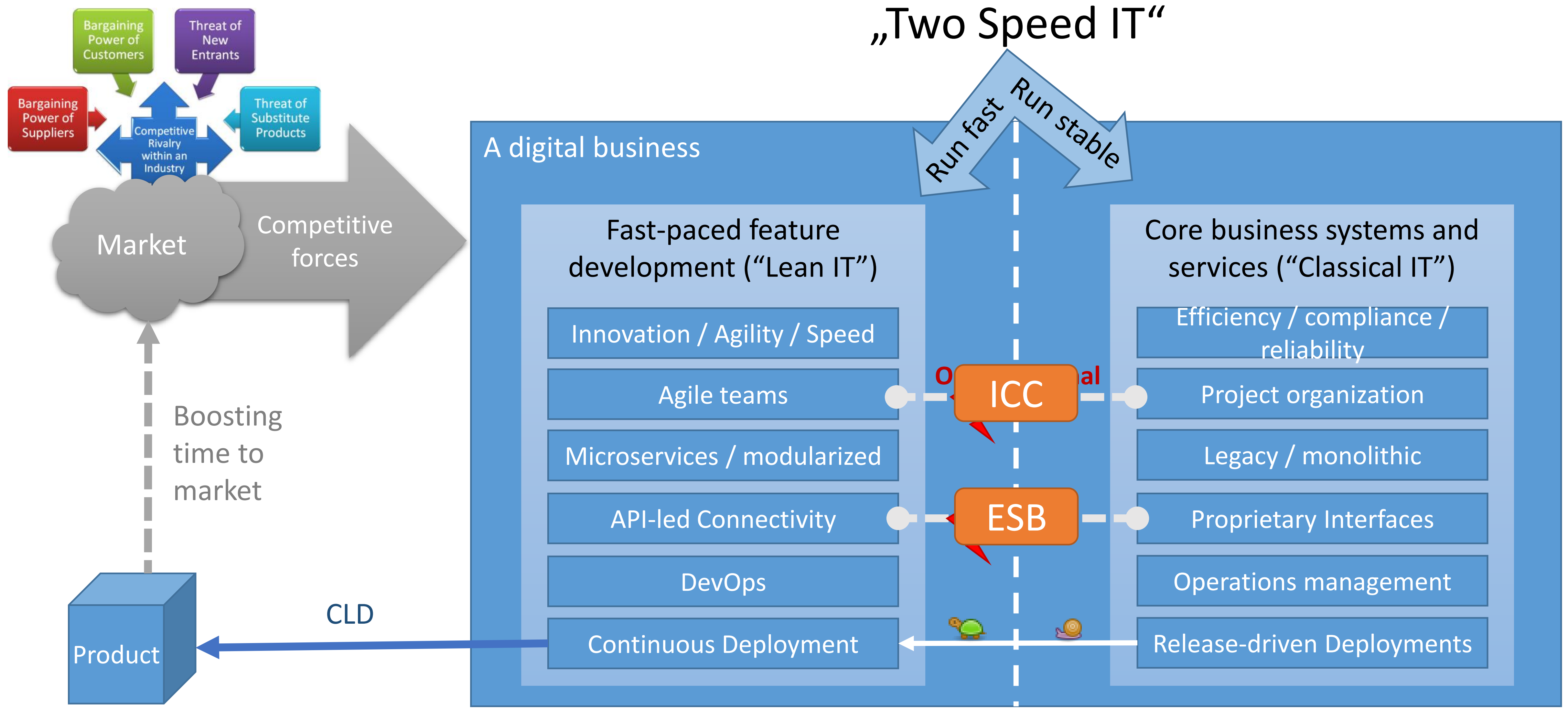
Competitive forces and what they do with Enterprise IT


Bargaining Power of Customers
Threat of New Entrants
Bargaining Power of Suppliers
Threat of Substitute Products
Competitive Rivalry within an Industry
Market
Competitive forces
Boosting time to market
Product
CLD
IT of an E-commerce business
Fast-paced feature development ("Lean IT")
Innovation / Agility / Speed
Agile teams
Microservices / modularized
API-led Connectivity
DevOps
Continuous Deployment
Core business systems and services ("Classical IT")
Efficiency / compliance / reliability
Project organization
Legacy / monolithic
Proprietary Interfaces
Operations management
Release-driven Deployments

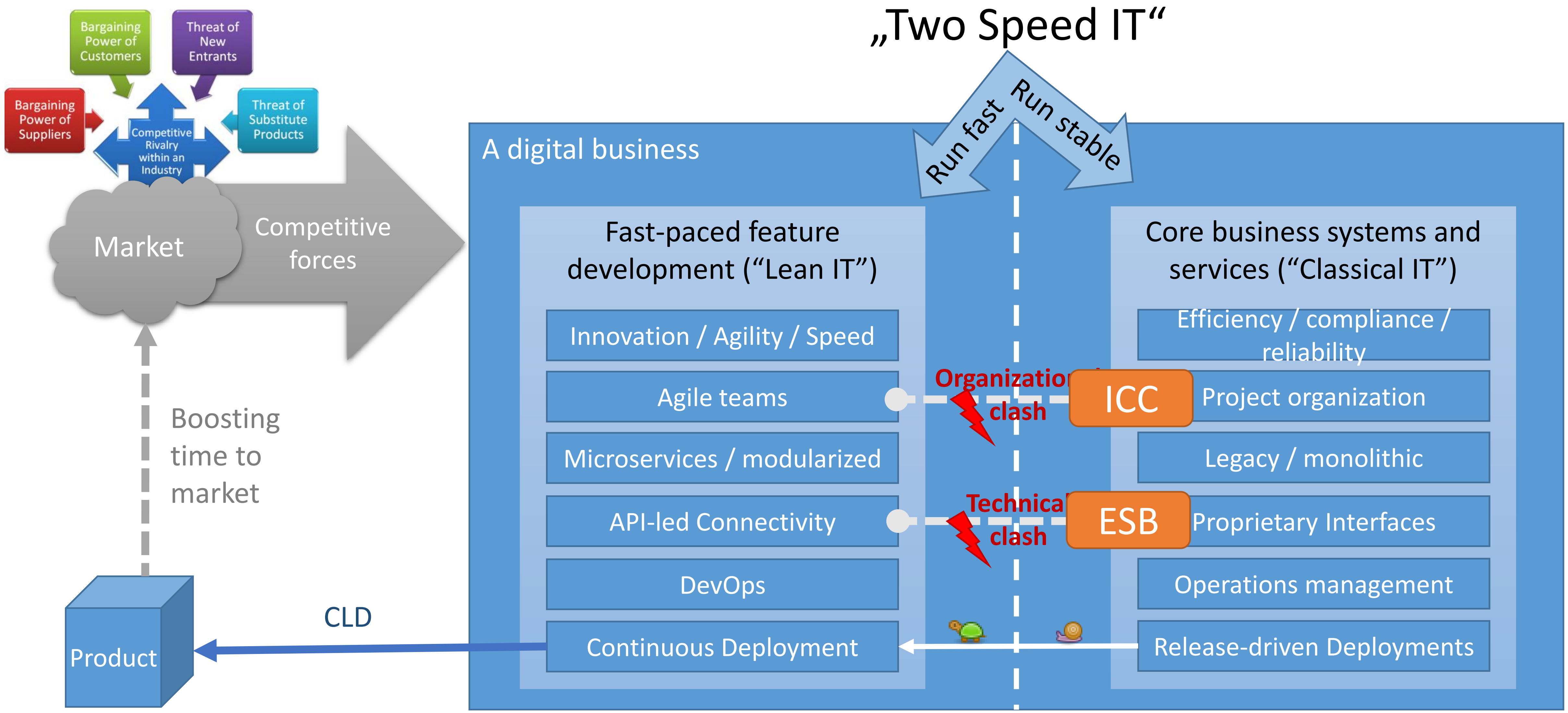
“Two speed IT” parts heavily depend on each other not only for the good


Bargaining Power of Customers
Threat of New Entrants
Bargaining Power of Suppliers
Threat of Substitute Products
Competitive Rivalry within an Industry
Market
Competitive forces
Boosting time to market
Product
CLD
„Two Speed IT“
Run fast
Run stable
A digital business
Fast-paced feature development (“Lean IT”)
Innovation / Agility / Speed
Agile teams
Microservices / modularized
API-led Connectivity
DevOps
Continuous Deployment
Organizational clash
Technical clash
Core business systems and services (“Classical IT”)
Efficiency / compliance / reliability
Project organization
Legacy / monolithic
Proprietary Interfaces
Operations management
Release-driven Deployments

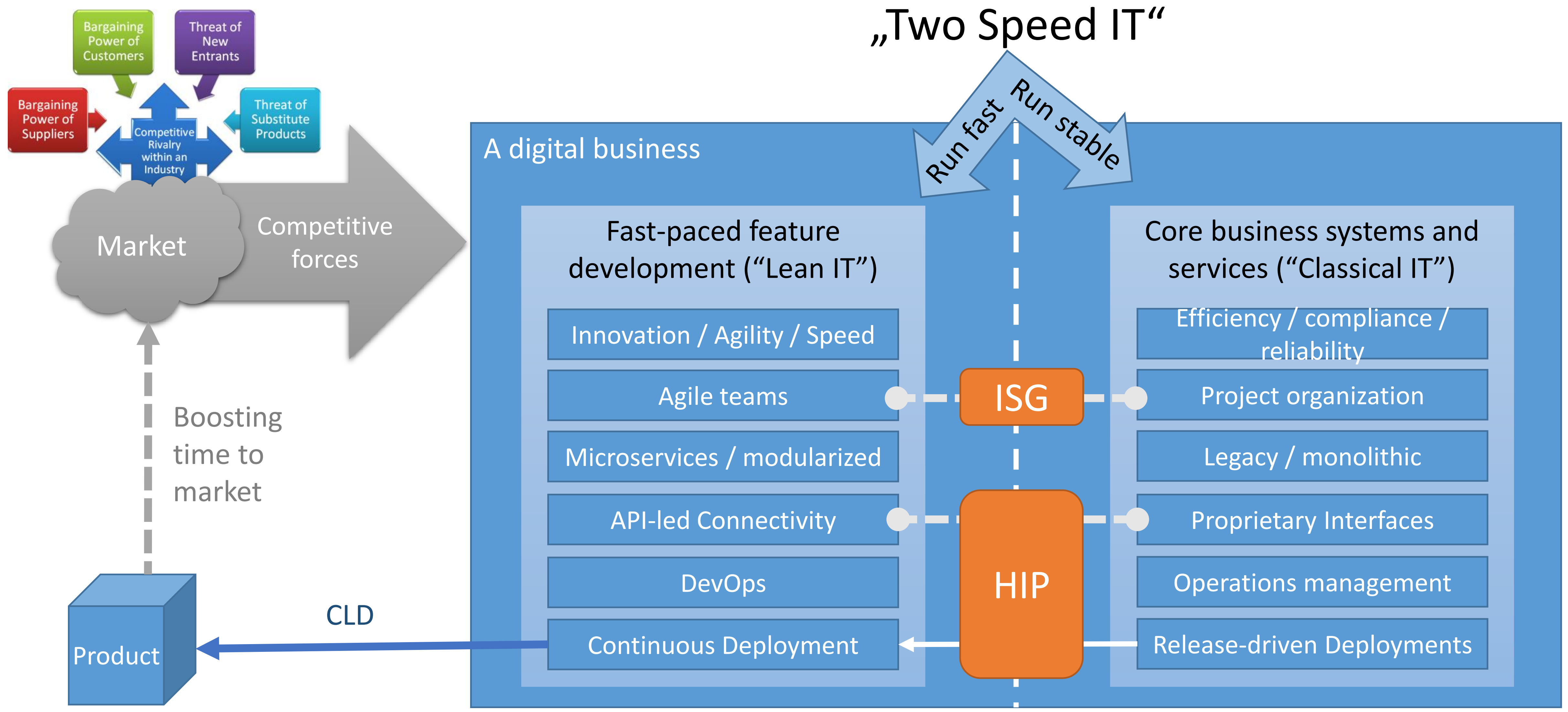
Overcome technical burden with Integration strategy (SOA?)


„Two Speed IT“
Run fast
Run stable
Bargaining Power of Customers
Threat of New Entrants
Bargaining Power of Suppliers
Threat of Substitute Products
Competitive Rivalry within an Industry
Market
Competitive forces
Boosting time to market
Product
CLD
A digital business
Fast-paced feature development (“Lean IT”)
Innovation / Agility / Speed
Agile teams
Microservices / modularized
API-led Connectivity
DevOps
Continuous Deployment
Core business systems and services (“Classical IT”)
Efficiency / compliance / reliability
Project organization
Legacy / monolithic
Proprietary Interfaces
Operations management
Release-driven Deployments
O
al
ICC
ESB

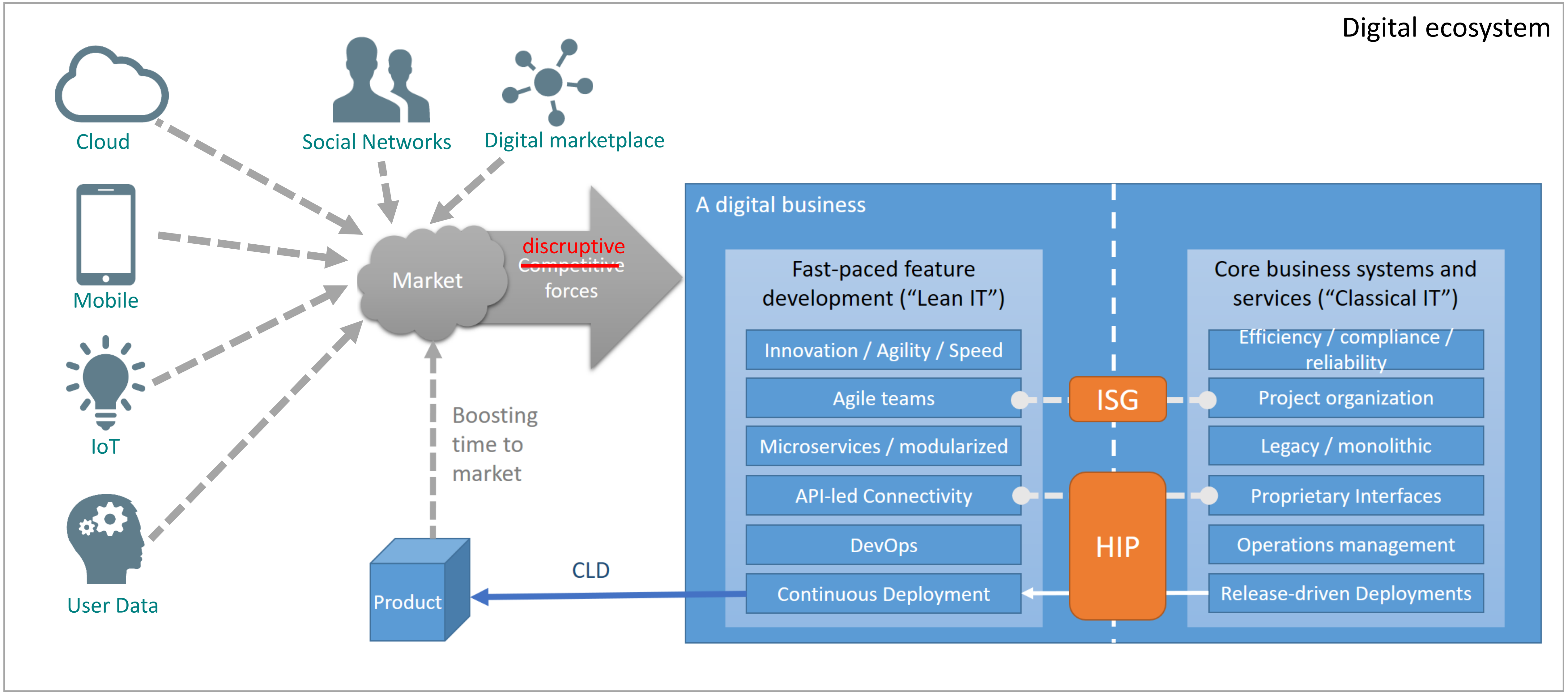
Well, not … Integration projects tend to be traditional (often) for good reason


Bargaining Power of Customers
Threat of New Entrants
Bargaining Power of Suppliers
Threat of Substitute Products
Competitive Rivalry within an Industry
Market
Competitive forces
Boosting time to market
Product
CLD
„Two Speed IT“
Run fast
Run stable
A digital business
Fast-paced feature development (“Lean IT”)
Innovation / Agility / Speed
Agile teams
Microservices / modularized
API-led Connectivity
DevOps
Continuous Deployment
Core business systems and services (“Classical IT”)
Efficiency / compliance / reliability
Project organization
Legacy / monolithic
Proprietary Interfaces
Operations management
Release-driven Deployments
Organization clash
ICC
Technical clash
ESB

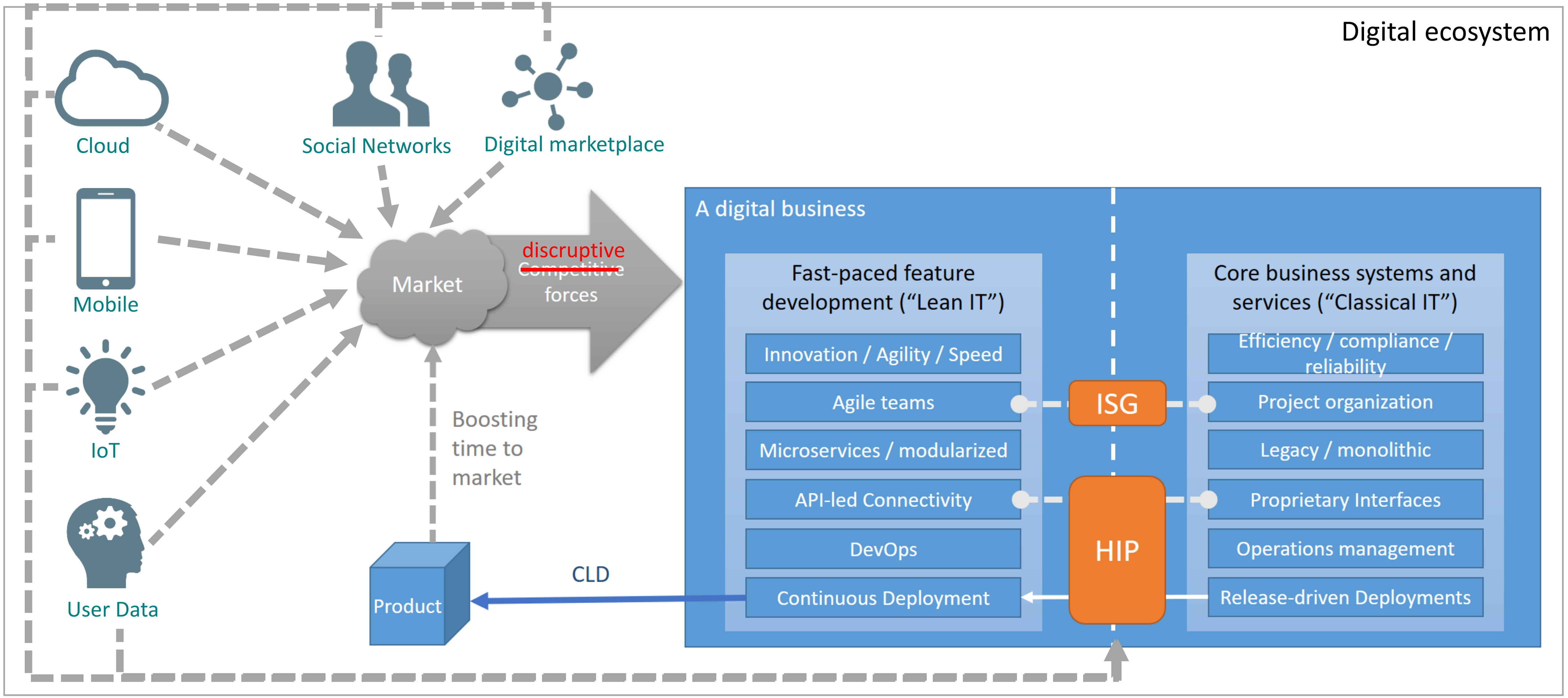
An integration strategy should follow a “bimodal” approach where so-called hybrid integration platforms (HIP) strive for Self Service Integration


Bargaining Power of Customers
Threat of New Entrants
Bargaining Power of Suppliers
Threat of Substitute Products
Competitive Rivalry within an Industry
Market
Competitive forces
Boosting time to market
Product
CLD
„Two Speed IT“
Run fast
Run stable
A digital business
Fast-paced feature development (“Lean IT”)
Innovation / Agility / Speed
Agile teams
Microservices / modularized
API-led Connectivity
DevOps
Continuous Deployment
ISG
HIP
Core business systems and services (“Classical IT”)
Efficiency / compliance / reliability
Project organization
Legacy / monolithic
Proprietary Interfaces
Operations management
Release-driven Deployments

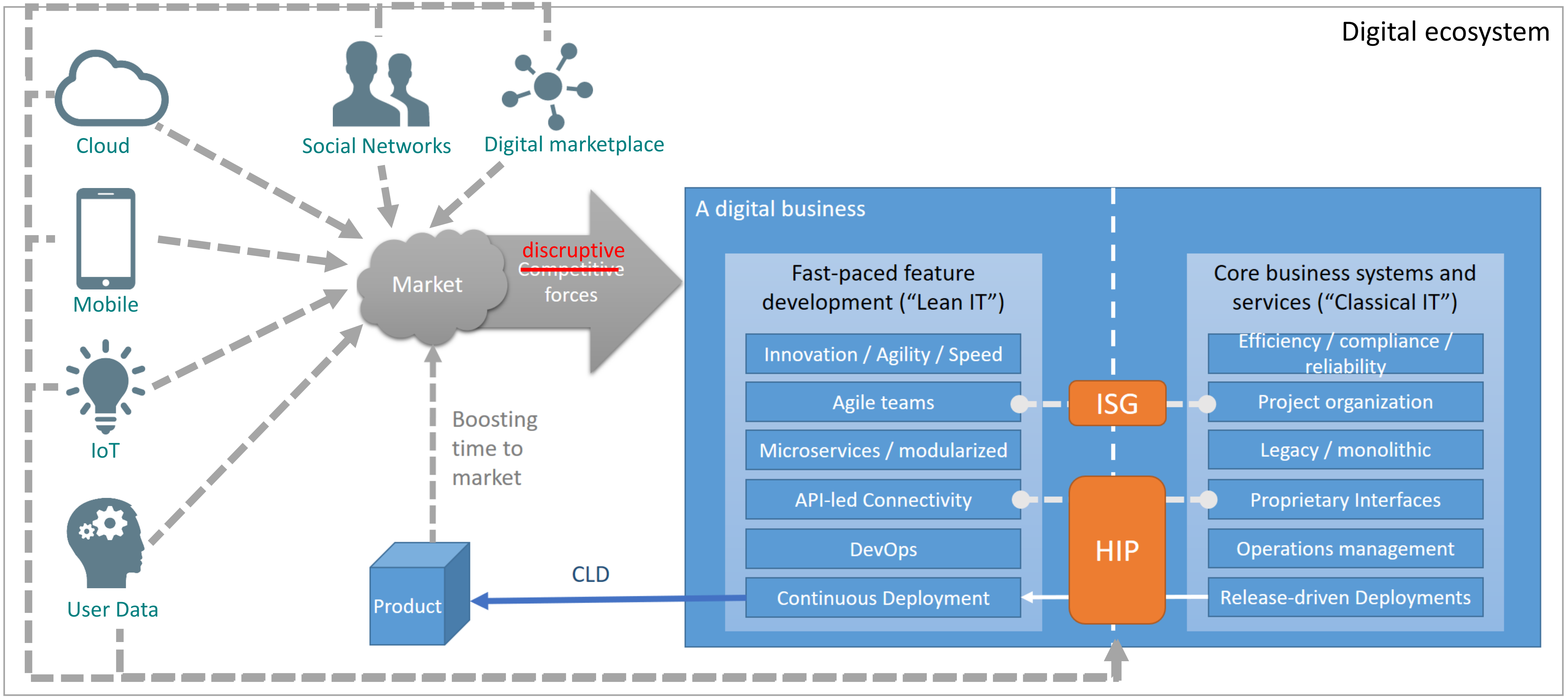
The digital ecosystem brings a lot of new potential but also threat



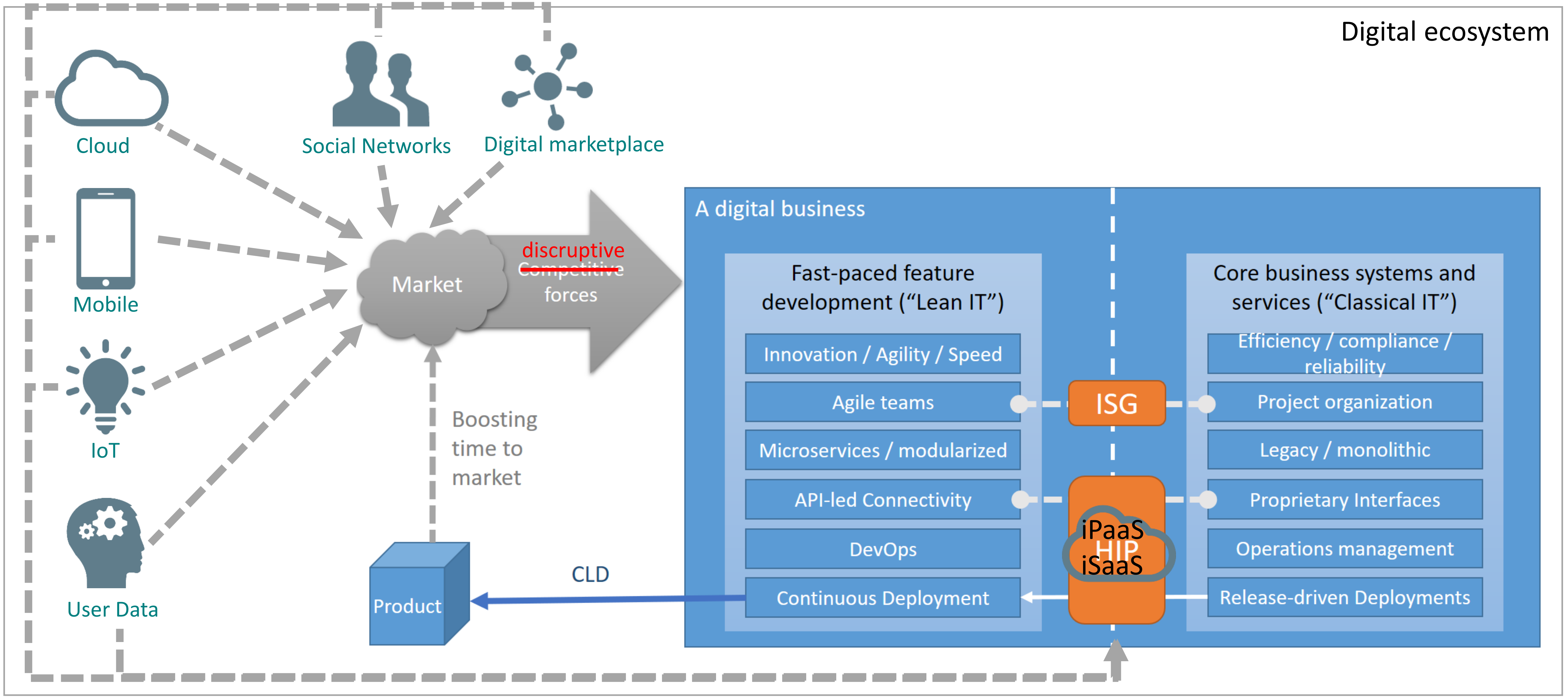
Businesses need to adopt and integrate these potential



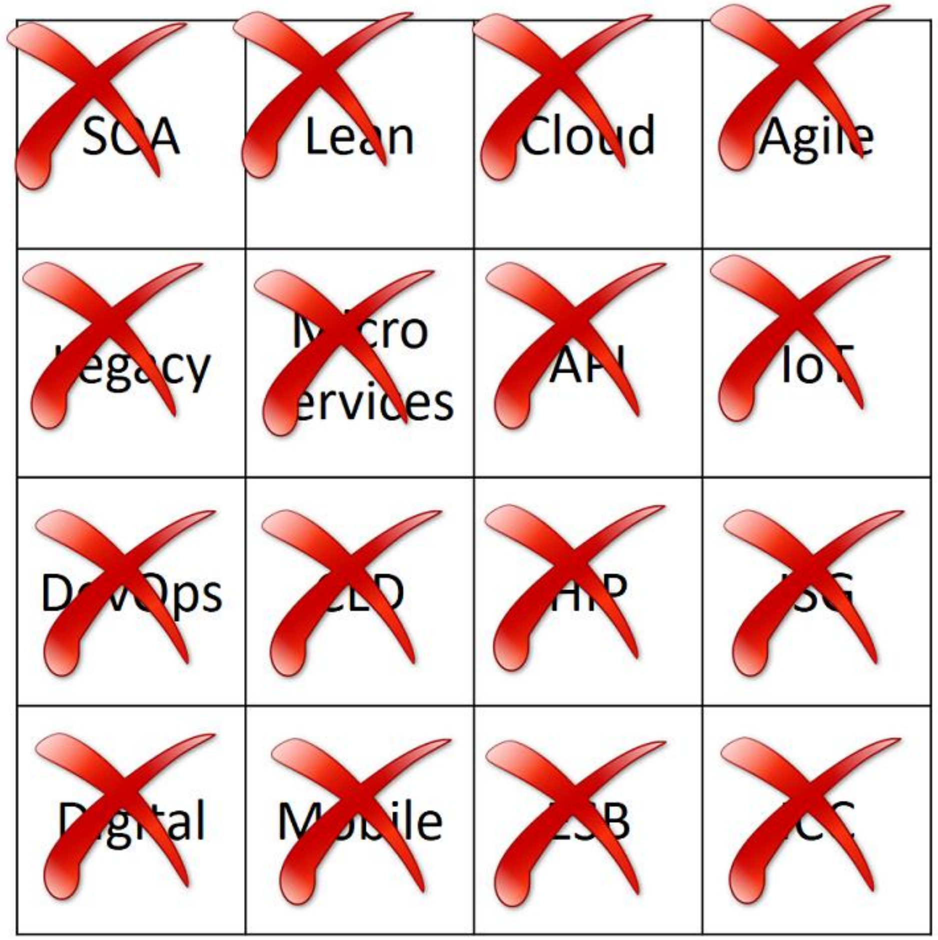
An HIP got its name from being the gateway for those new stuff



HIP is a concept whereas iPaaS, iSaaS are (commercial) solutions in the cloud



Ok, you got your BINGO! Let’s move on in the real world …



SOA	Lean	Cloud	Agile
Legacy	Micro Services	API	IoT
DevOps	CLD	HTP	5G
Digital	Mobile	ESB	ICC

Agenda

- Disruptive forces and what they do with enterprise IT
- **An ideal integration platform**
- AWS Simple Workflows (SWF) in a nutshell
- Demo time
- Leveraging SWF to get rid of a classical ESB solution
- Reclaim process ownership and end-2-end-autonomy
- Drawing the big picture of a hybrid integration solution

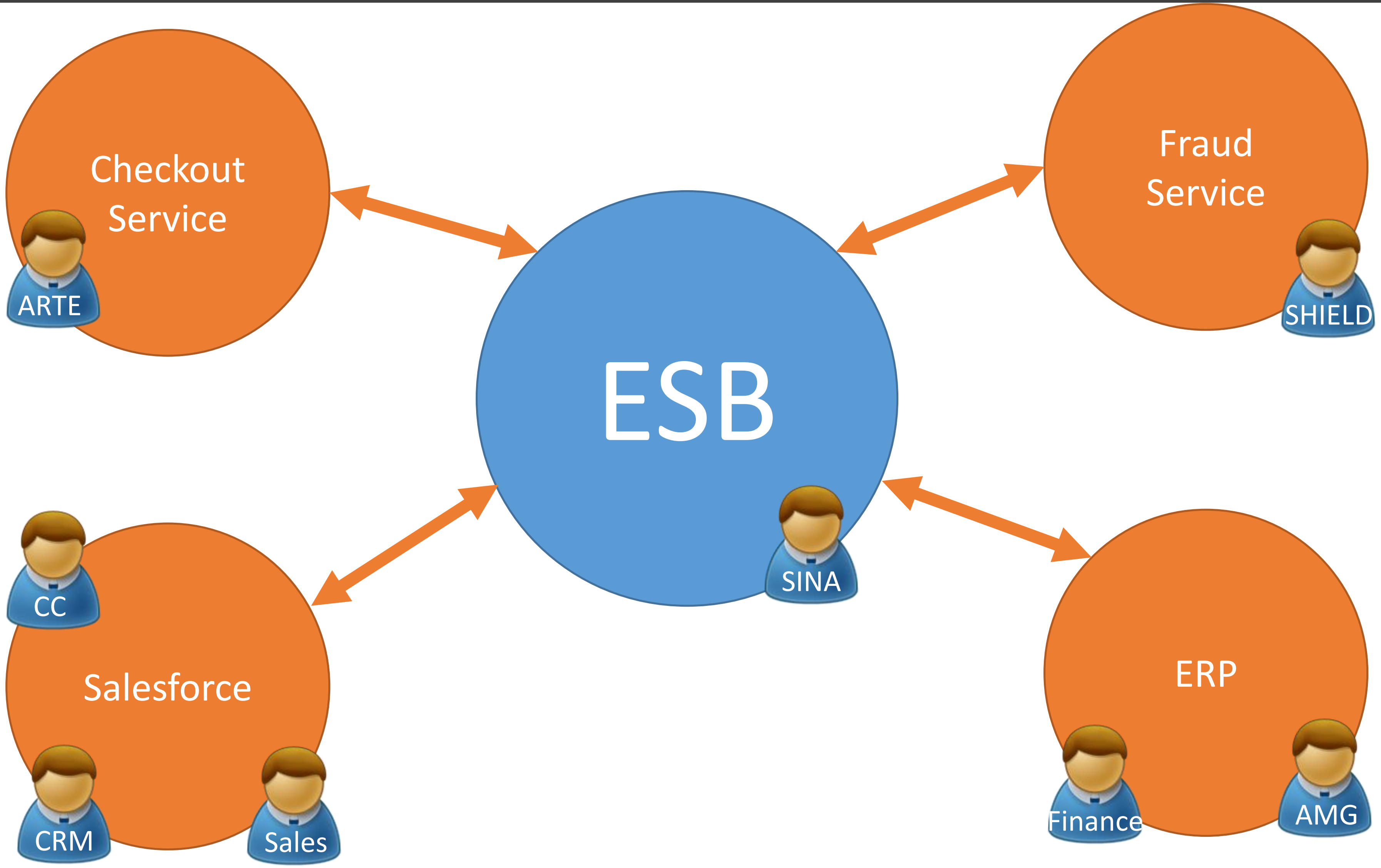
Given an ESB in charge of syncing business data across the enterprise

Customer
Contract
Article

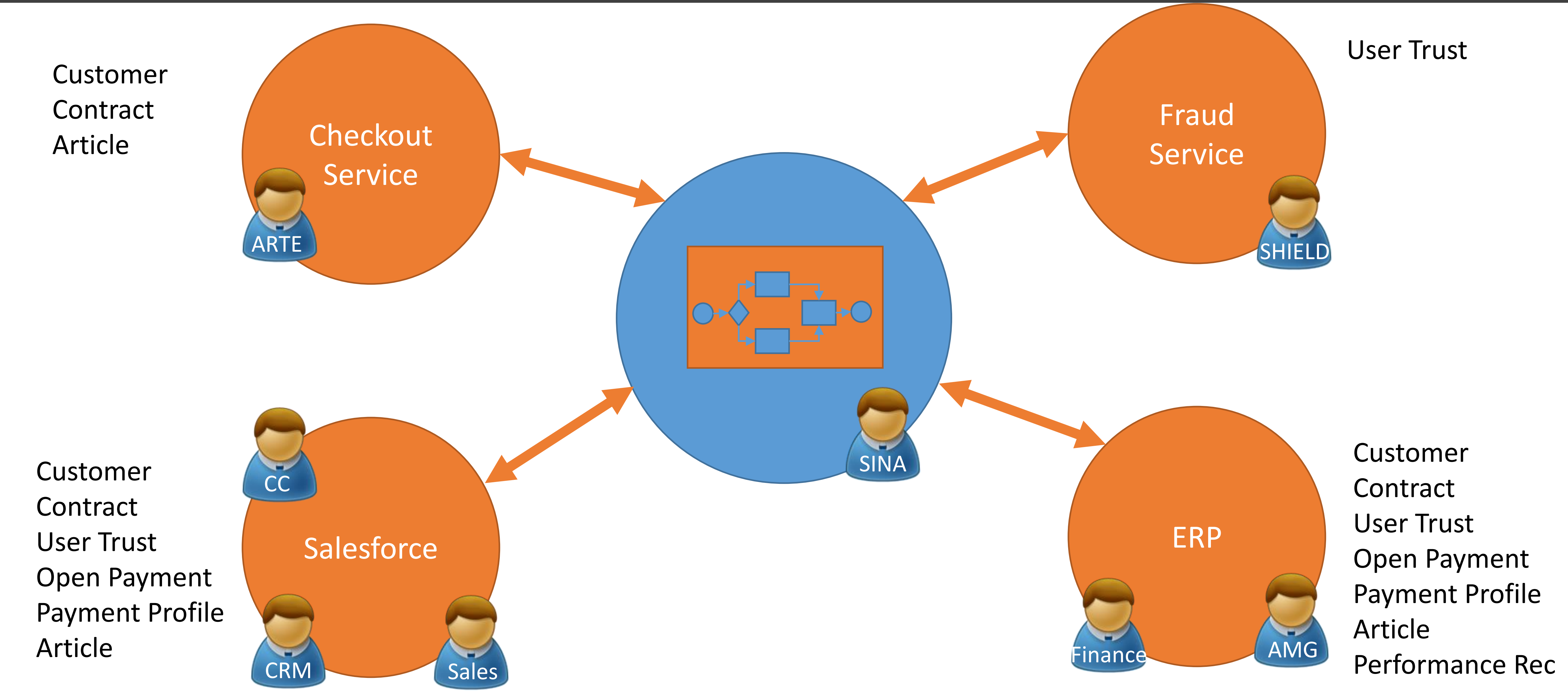
User Trust

Customer
Contract
User Trust
Open Payment
Payment Profile
Article

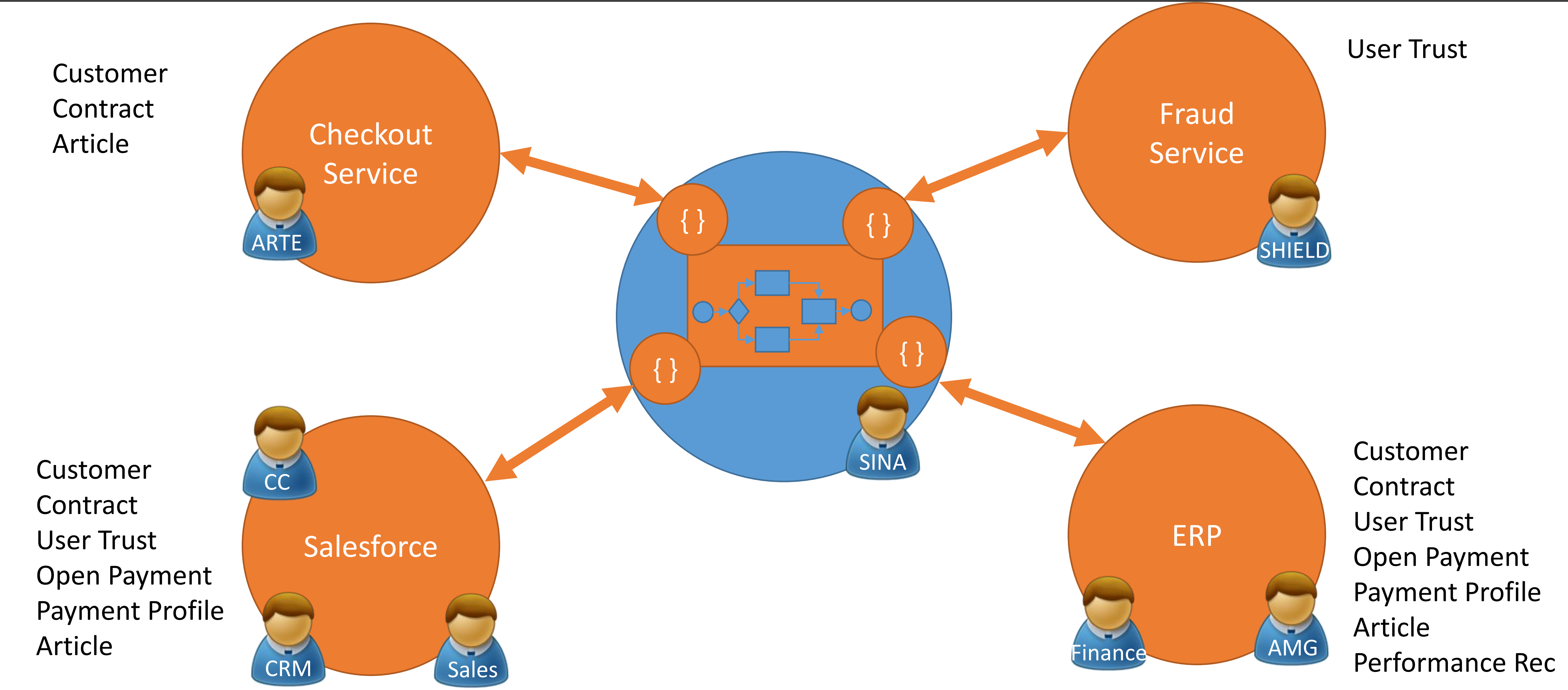
Customer
Contract
User Trust
Open Payment
Payment Profile
Article
Performance Rec



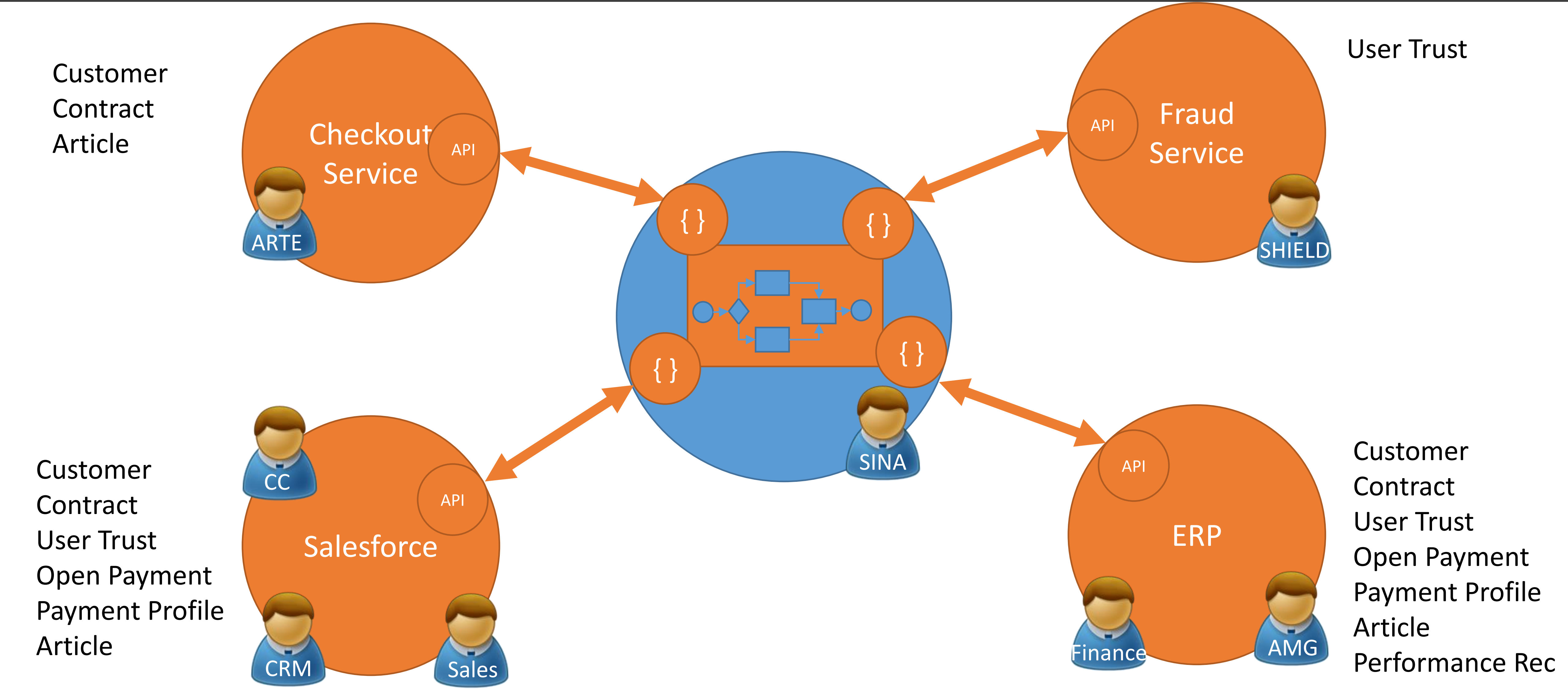
This centralized hub is key for processes arching over multiple domains



It encapsulates the burden of integrating system interfaces with brokers



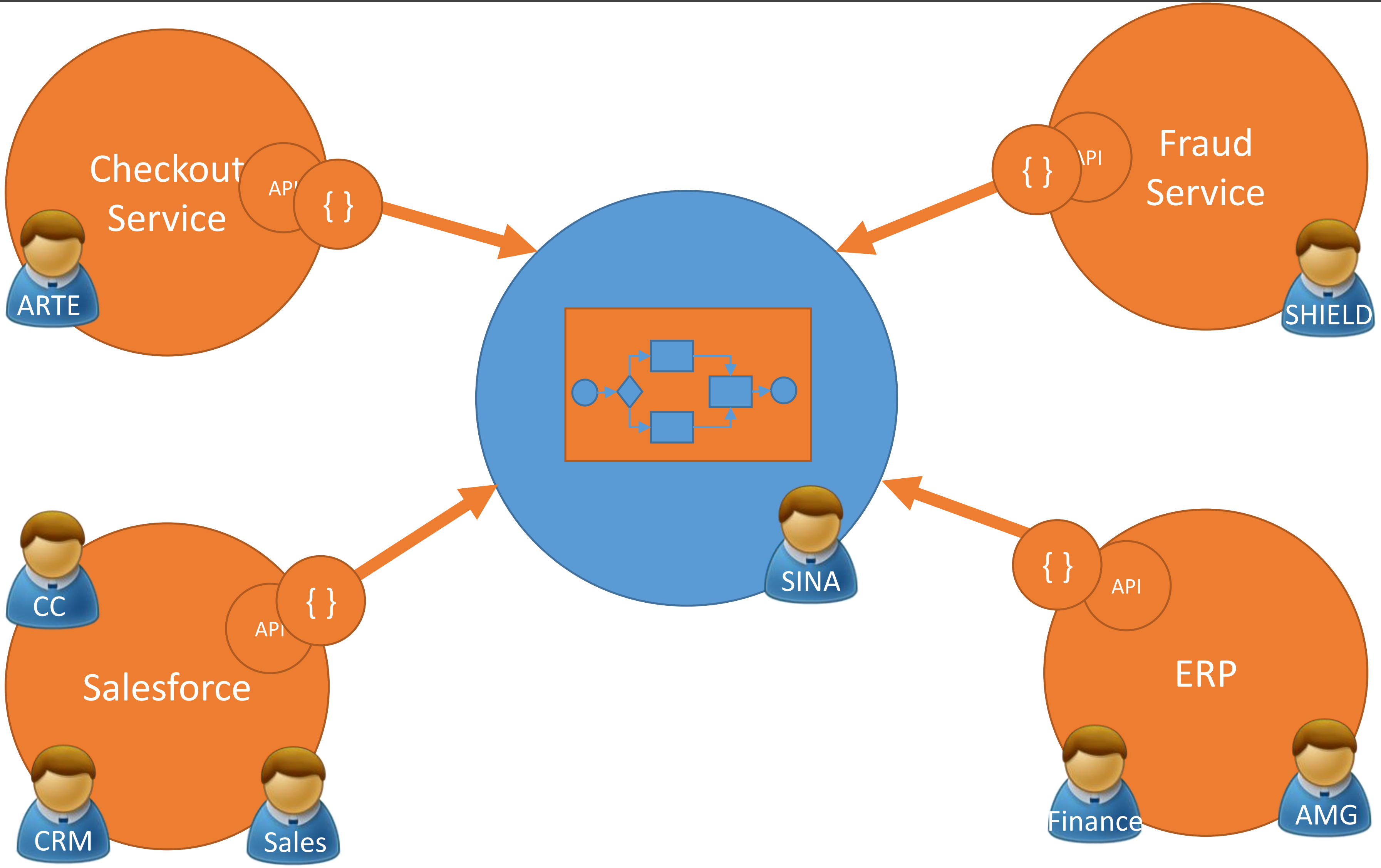
Teams depend on those brokers in order to change their interfaces



It would be great to push responsibility (integration logic) to the edges

Customer
Contract
Article

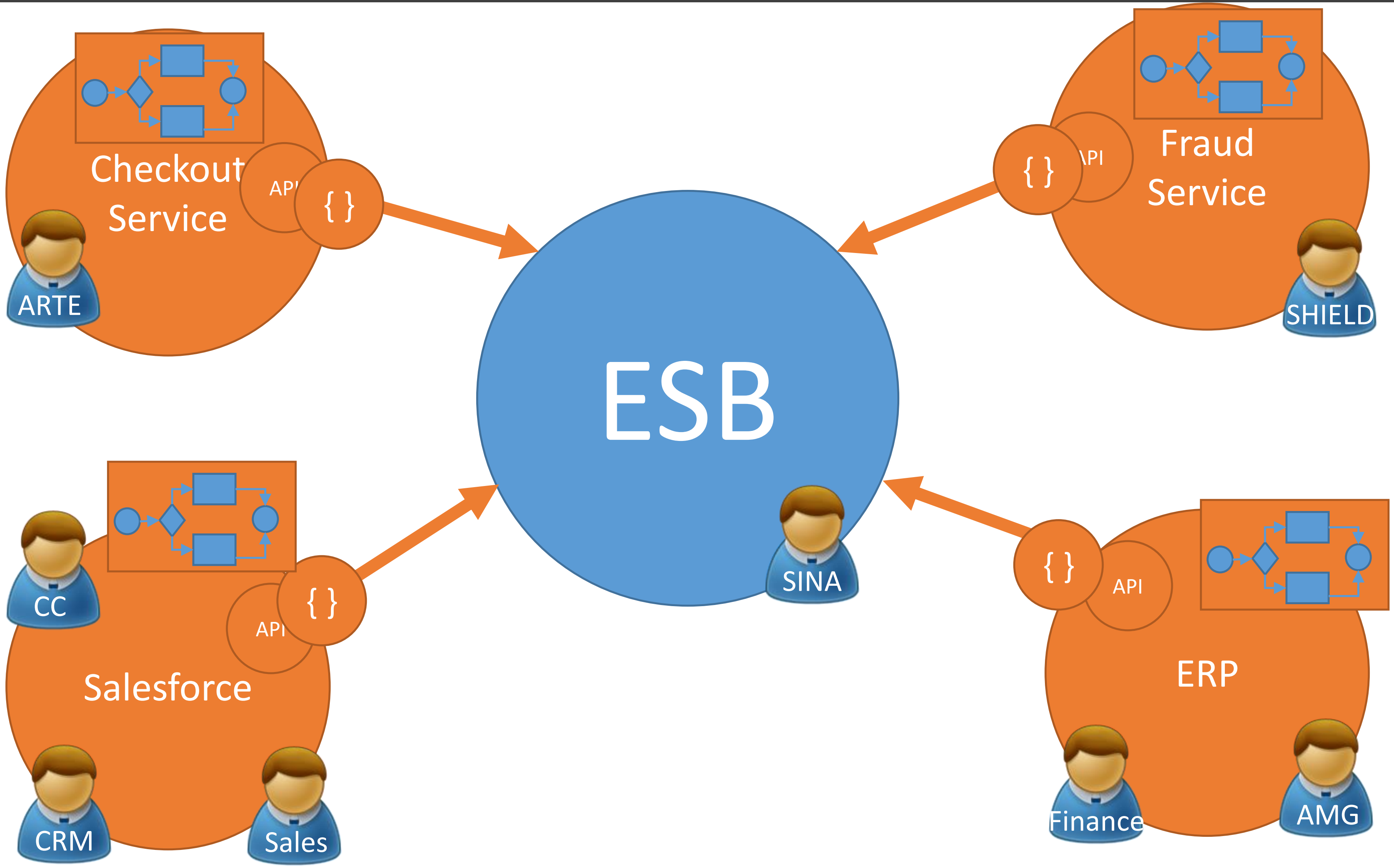
User Trust



Customer
Contract
User Trust
Open Payment
Payment Profile
Article

Customer
Contract
User Trust
Open Payment
Payment Profile
Article
Performance Rec

It would also be great to delegate process ownership (orchestration logic)



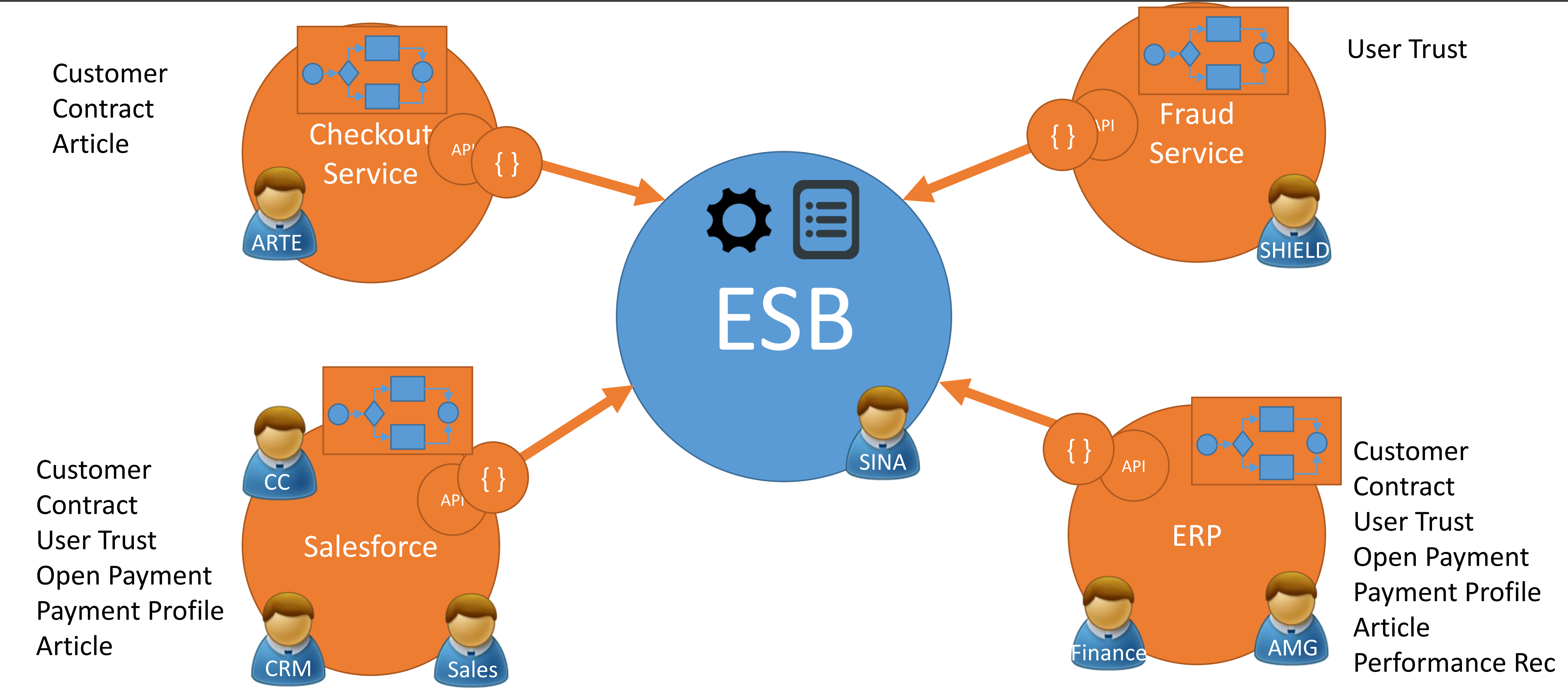
Customer
Contract
Article

User Trust

Customer
Contract
User Trust
Open Payment
Payment Profile
Article

Customer
Contract
User Trust
Open Payment
Payment Profile
Article
Performance Rec

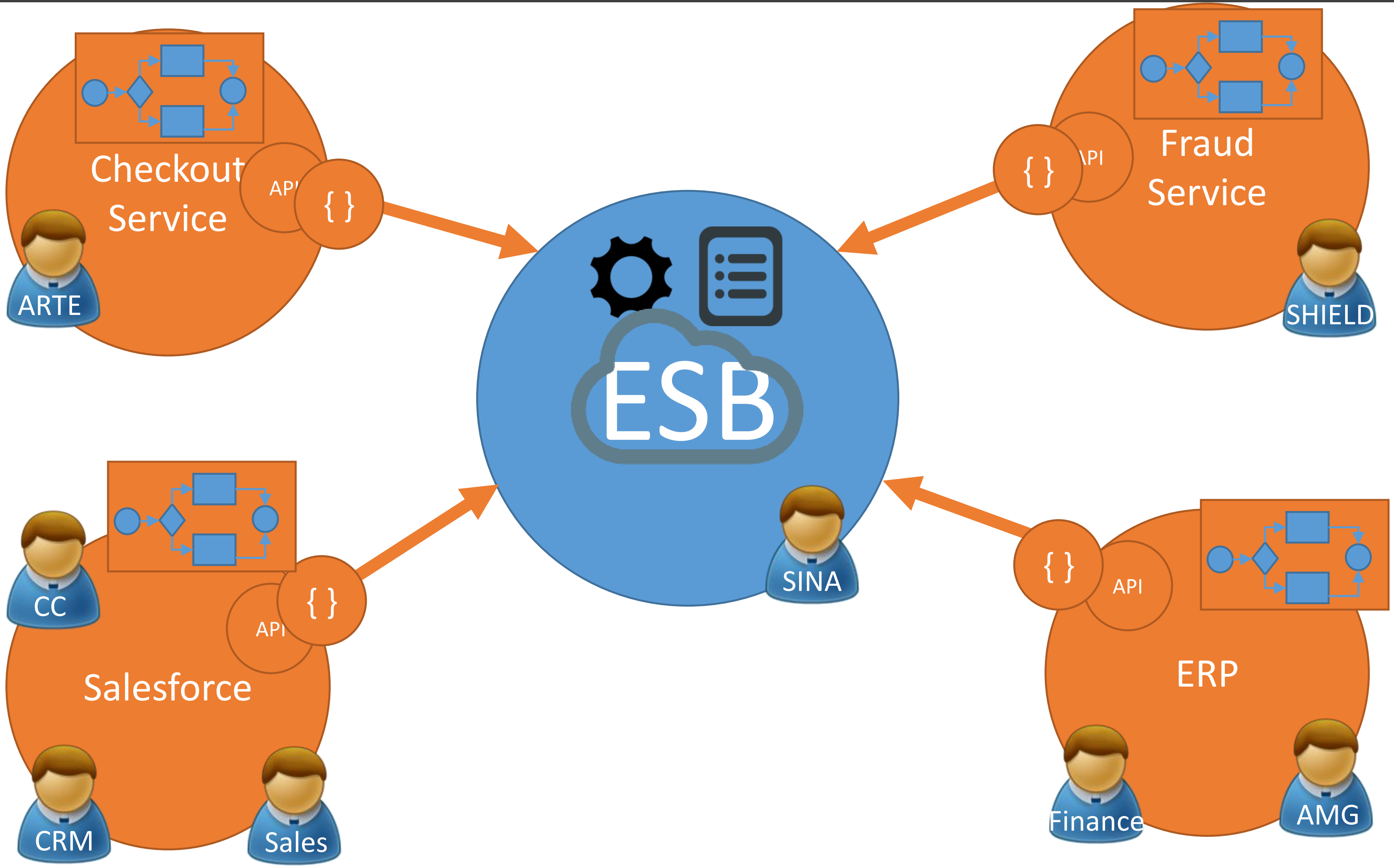
ESB left over for “dirty work” (messaging, tracking, governance, ...)



It should also be in the cloud to not be isolated from the digital ecosystem

Customer
Contract
Article

User Trust



Customer
Contract
User Trust
Open Payment
Payment Profile
Article

Customer
Contract
User Trust
Open Payment
Payment Profile
Article
Performance Rec

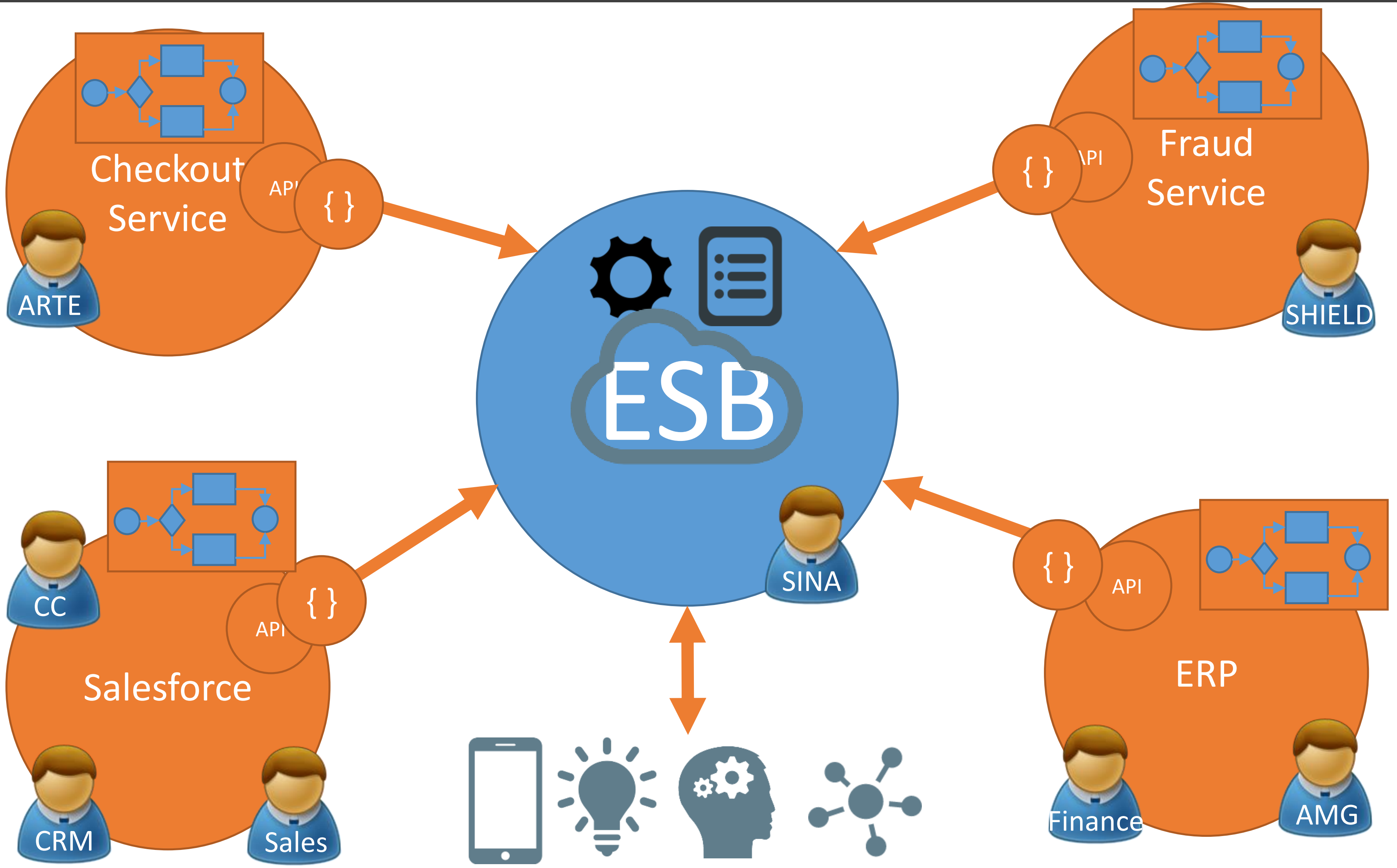
It should also be in the cloud to not be isolated from the digital ecosystem

Customer
Contract
Article

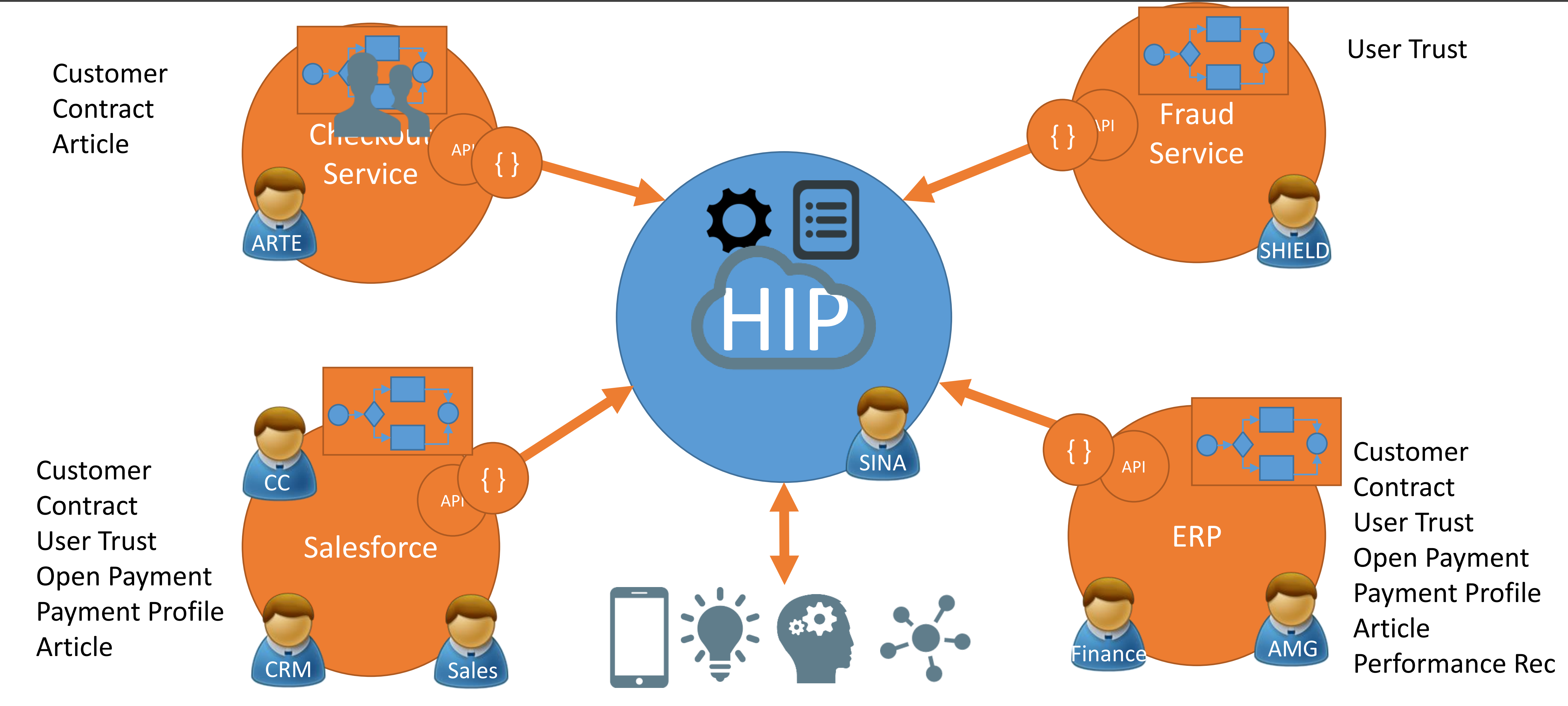
User Trust

Customer
Contract
User Trust
Open Payment
Payment Profile
Article

Customer
Contract
User Trust
Open Payment
Payment Profile
Article
Performance Rec

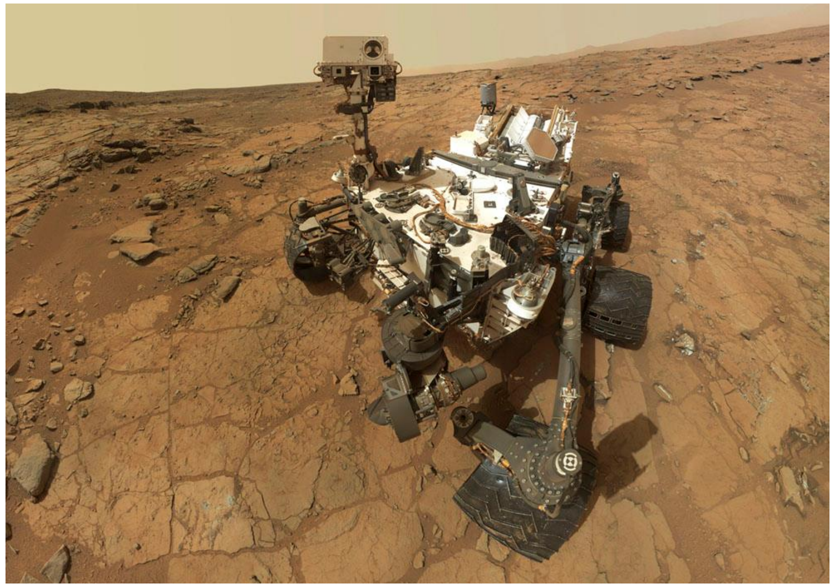


Why can't this thing just be like an HIP?



As our company moves to AWS anyway, let’s give SWF a try

- There is PAYG
- AWS ecosystem is huge, it serves all the hip stuff like serverless architecture (Lambda, API Gateway), IoT, Mobile integration, Messaging, Elastic computing, Container deployments and more
- There‘s a big community around AWS
- SWF (Simple Workflows) is used by NASA for processing data from the Mars-Rover on earth

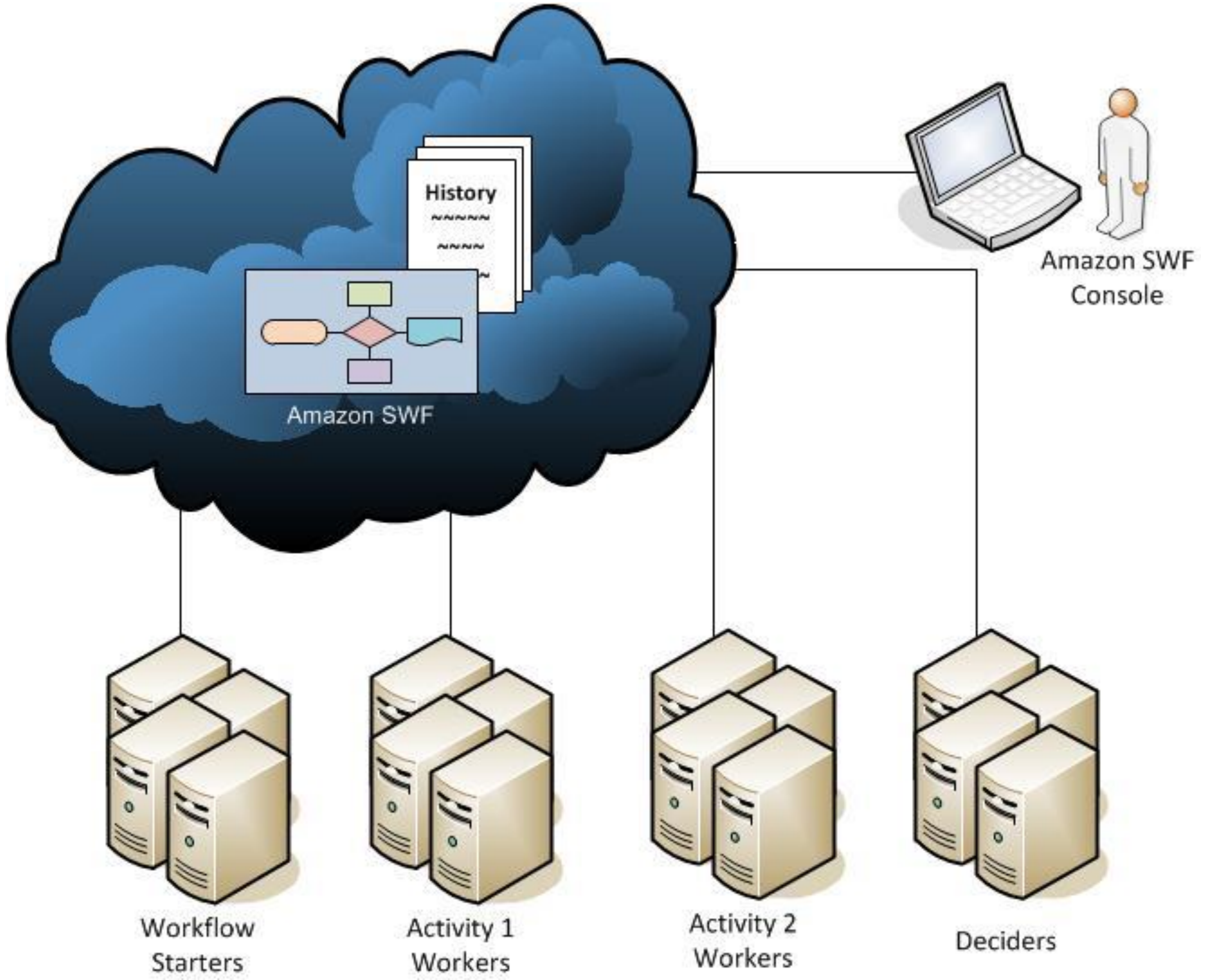


You got me at „Mars“…

Agenda

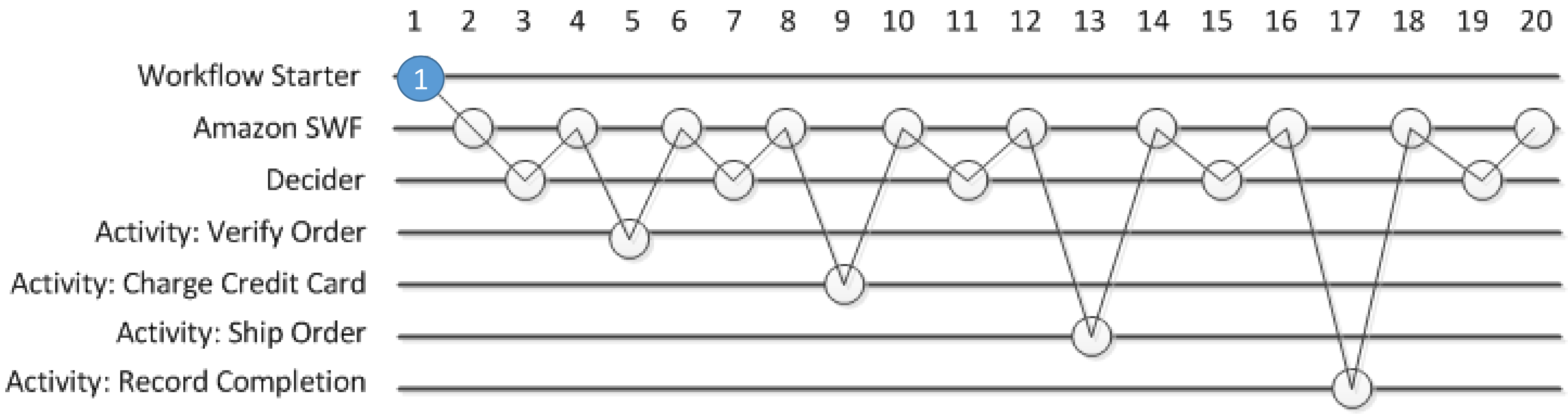
- Disruptive forces and what they do with enterprise IT
- An ideal integration platform
- **AWS Simple Workflows (SWF) in a nutshell**
- Demo time
- Leveraging SWF to get rid of a classical ESB solution
- Reclaim process ownership and end-2-end-autonomy

SWF is a workflow engine scheduling tasks for all the workflow participants



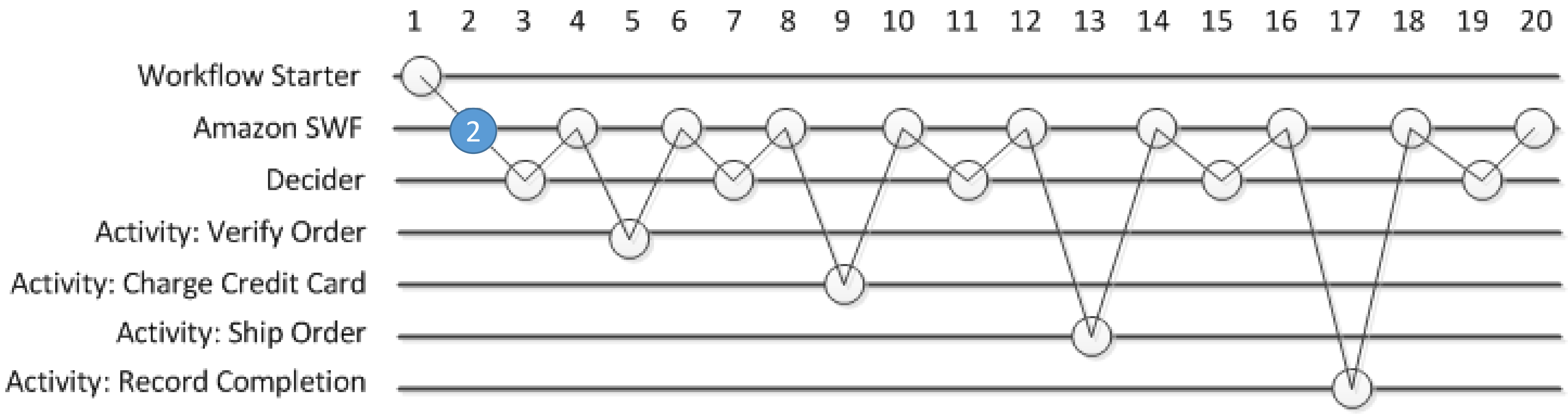
A workflow starter simply kicks off a workflow with some input via API

```
{ "childPolicy": "string", "domain": "string", "executionStartToCloseTimeout": "string", "input": "string", "lambdaRole": "string", "tagList": [
"string" ], "taskList": { "name": "string" }, "taskPriority": "string", "taskStartToCloseTimeout": "string", "workflowId": "string", "workflowType":
{ "name": "string", "version": "string" } }
```



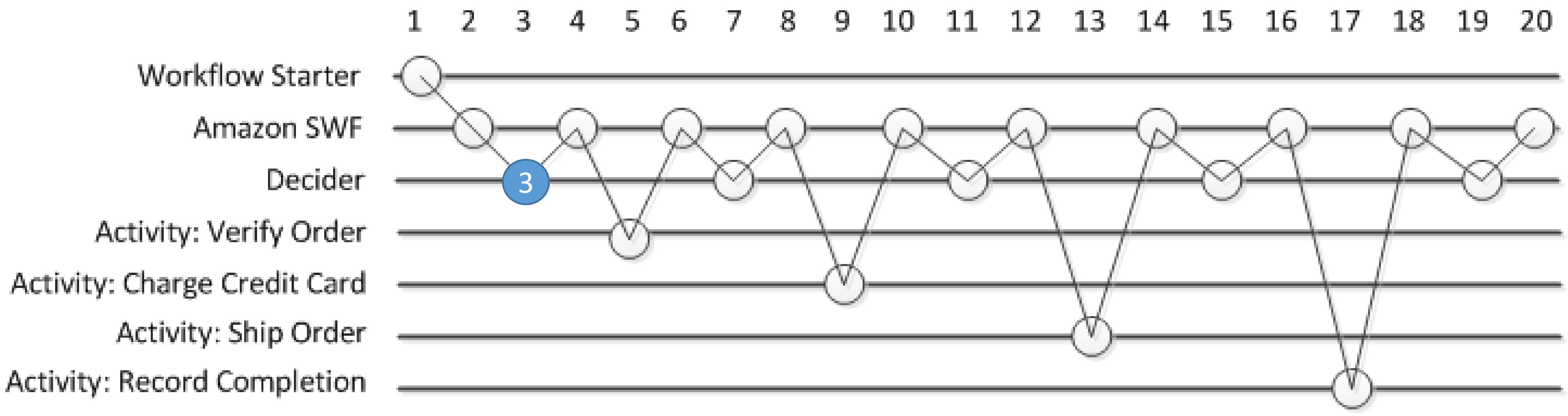
```
{ "runId": "string" }
```

SWF has no clue what comes next so it schedules a “decision task”



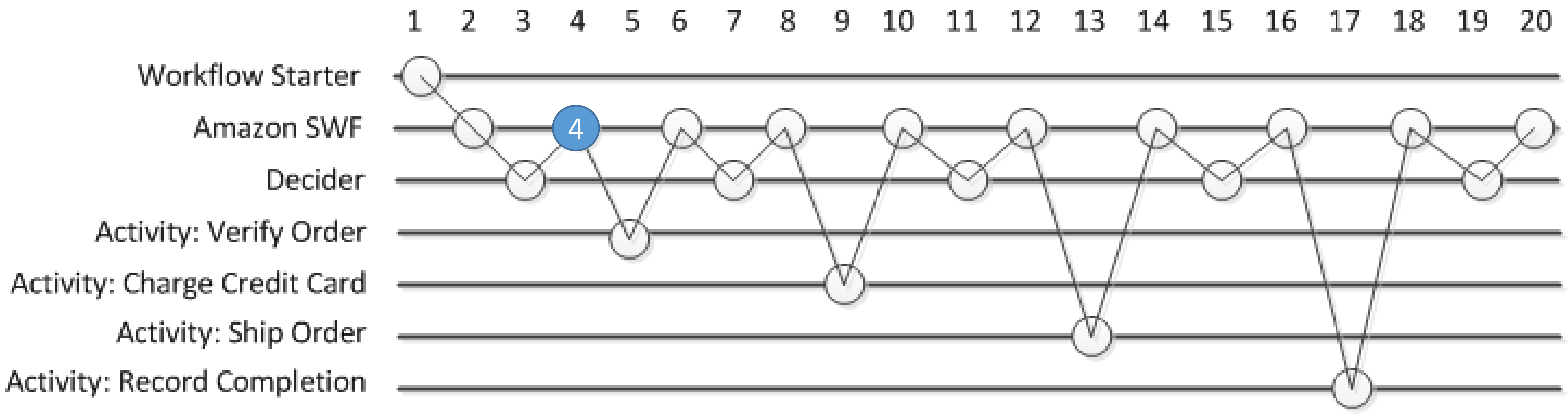
A decider owns the actual workflow logic. It is an application polling for tasks via API and returns decisions to the SWF engine

```
{ "domain": "string", "identity": "string", "maximumPageSize": number, "nextPageToken": "string", "reverseOrder": boolean, "taskList": { "name":
"string" } }
```



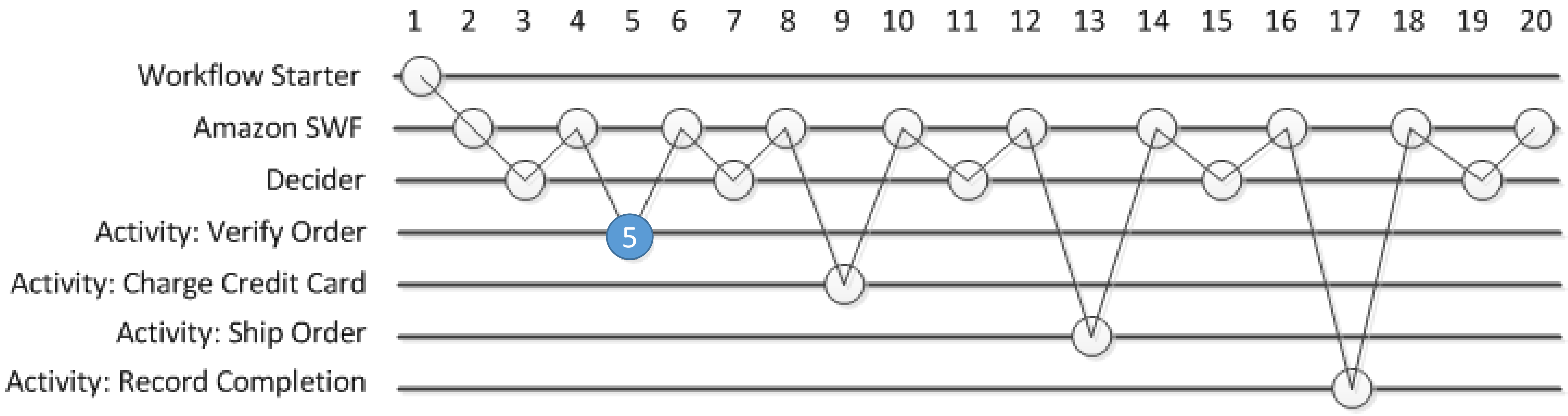
```
{ "decisions": [ { "cancelTimerDecisionAttributes": { … }, "cancelWorkflowExecutionDecisionAttributes": { … },
"completeWorkflowExecutionDecisionAttributes": { … }, "continueAsNewWorkflowExecutionDecisionAttributes": { … }, "recordMarkerDecisionAttributes": {
… }, "requestCancelActivityTaskDecisionAttributes": { … }, "requestCancelExternalWorkflowExecutionDecisionAttributes": { … },
"scheduleActivityTaskDecisionAttributes": { … }, "scheduleLambdaFunctionDecisionAttributes": { … },
"signalExternalWorkflowExecutionDecisionAttributes": { … }, "startChildWorkflowExecutionDecisionAttributes": { … }, "startTimerDecisionAttributes":
{ … } } ], "executionContext": "string", "taskToken": "string" }
```

SWF fulfills the remotely given decisions –> e.g. it schedules an activity task



An activity worker polls for tasks from a task list, works on it and returns a result

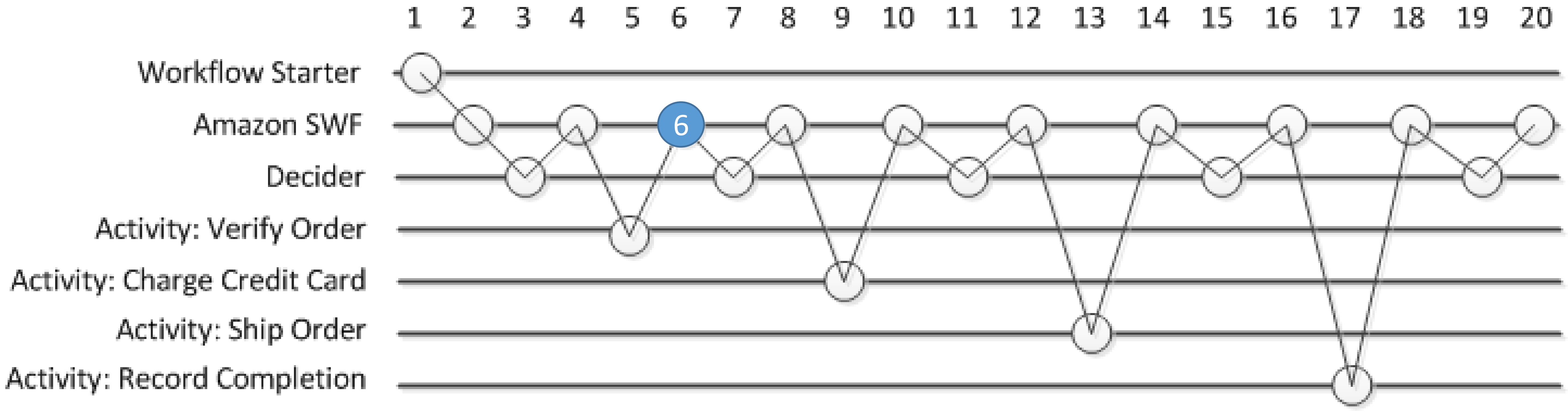
```
{ "domain": "string", "identity": "string", "taskList": { "name": "string" } }
```



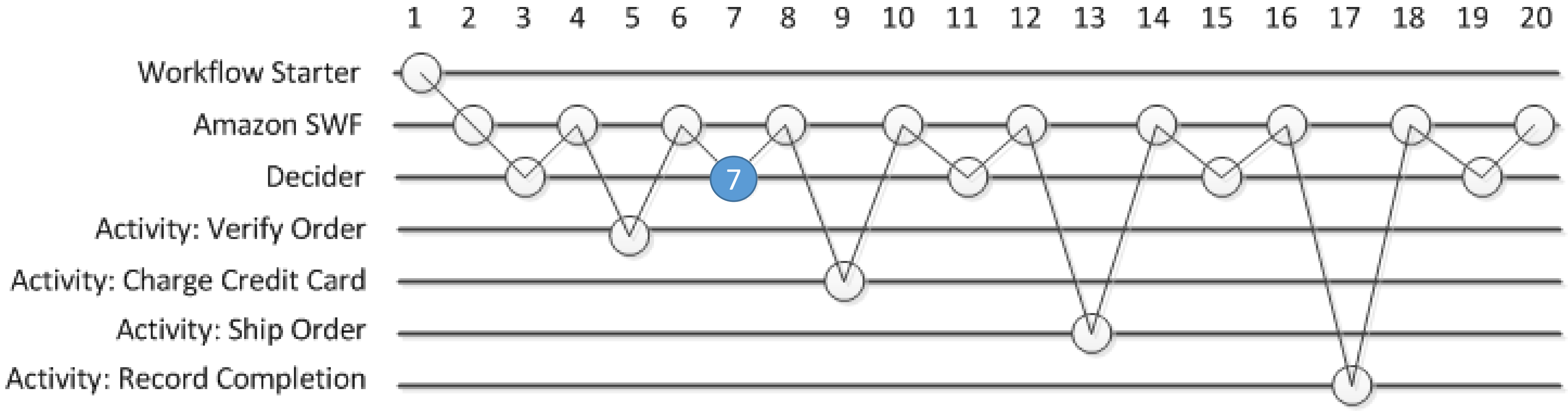
```
{ "result": "string", "taskToken": "string" }
```

```
{ "details": "string", "reason": "string", "taskToken": "string" }
```

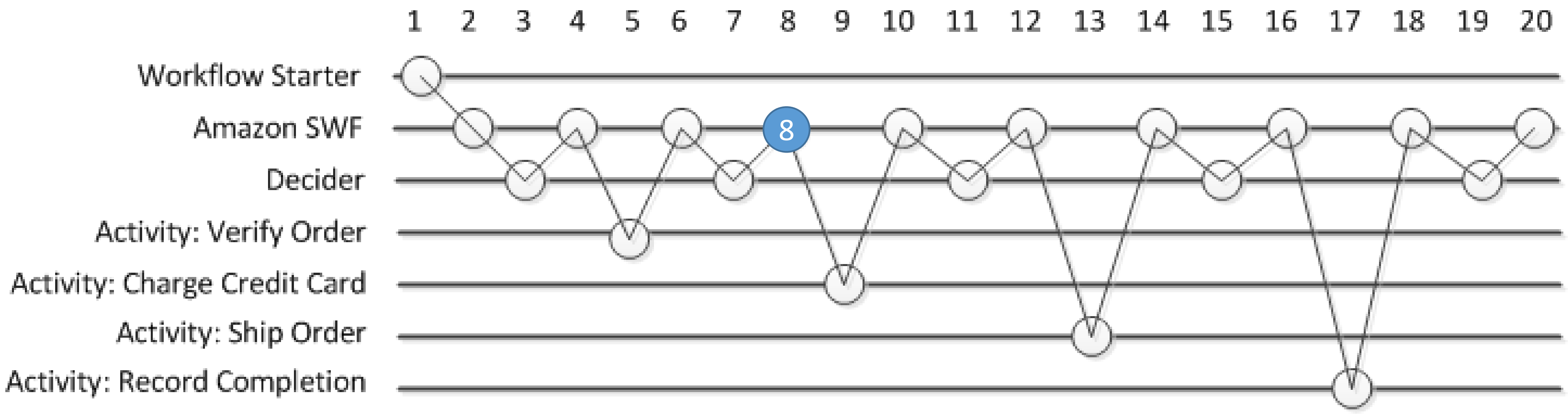
SWF receives the result – again has no clue how to go on – so it schedules another decision task



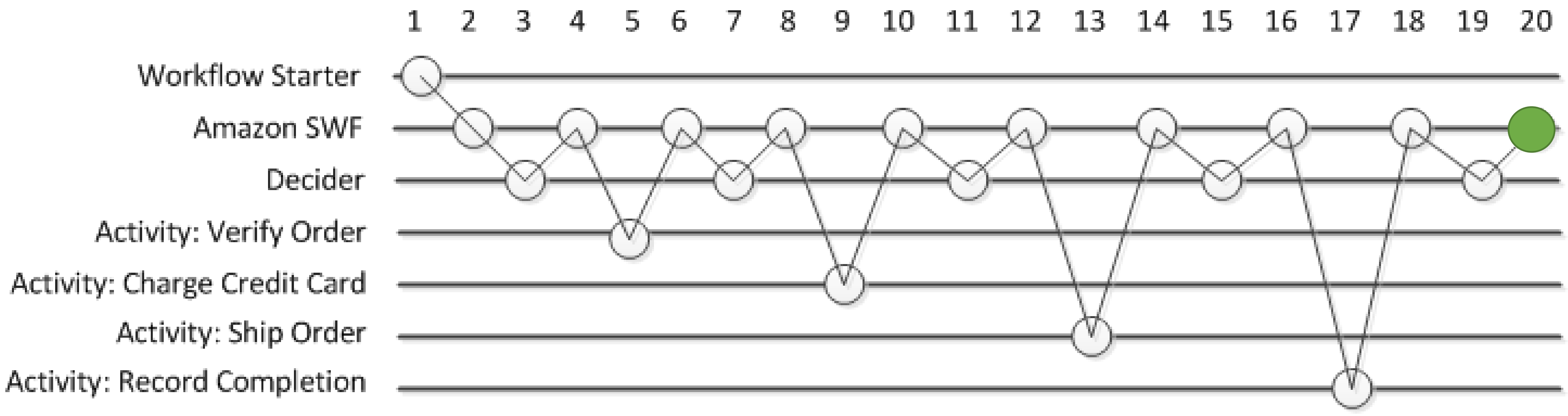
The decider receives the task. It now contains a detailed protocol of what happened in this workflow-execution. Based on that it gives the next decision(s)



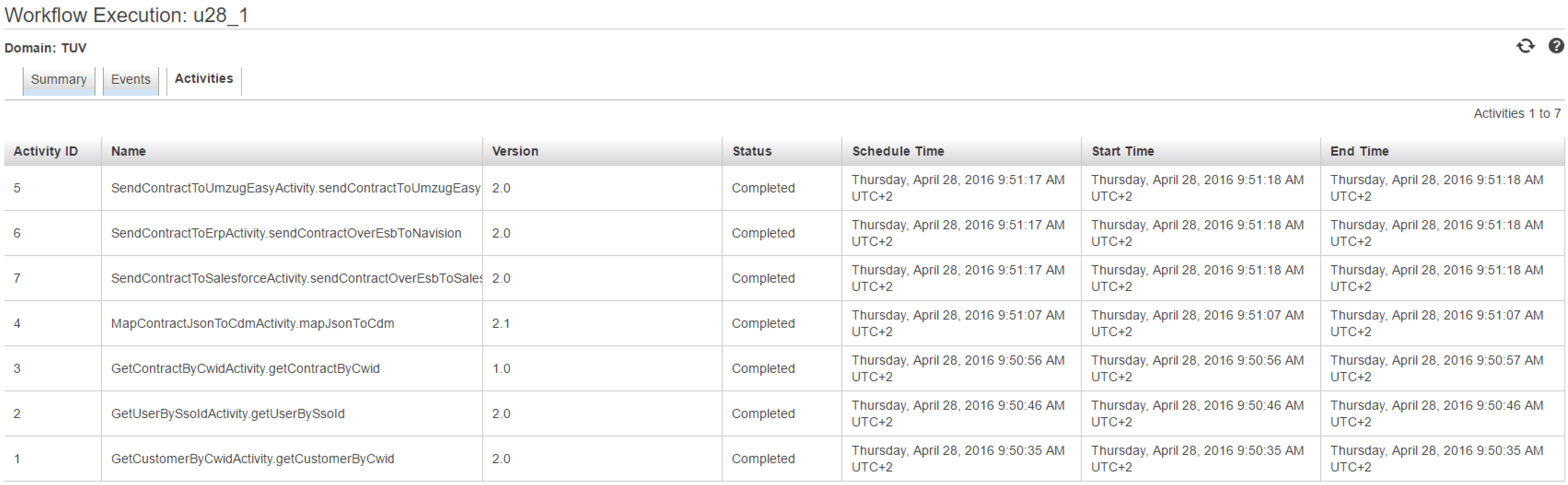
SWF schedules the next task – according to the last decision by the decider. Activity workers can be serverless Lambda-functions as well



This goes on and on until the decider aka workflow worker decides for completing the workflow. What again is executed by SWF itself.



You keep track of your workflow executions in AWS console



Workflow Execution: u28_1

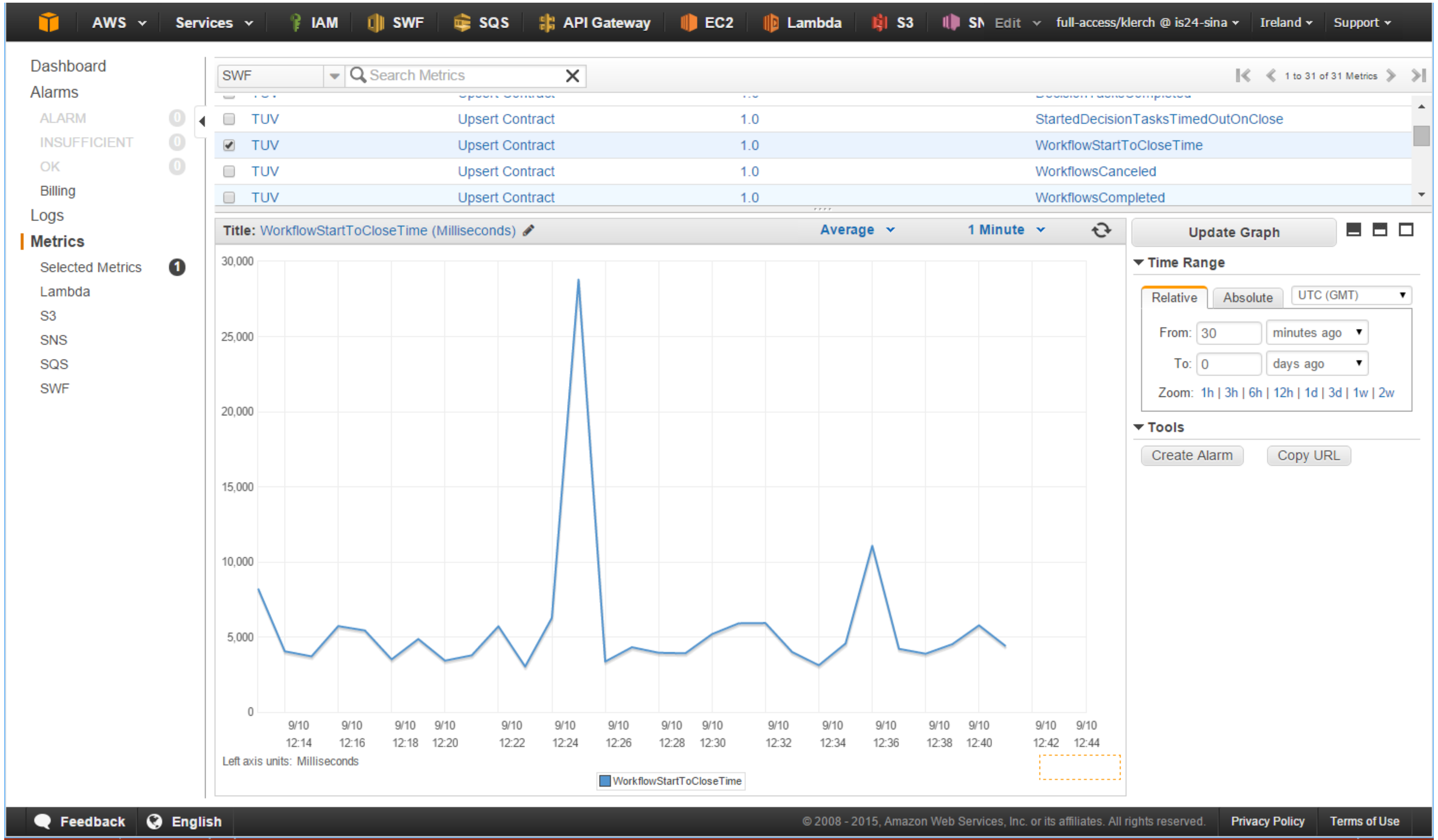
Domain: TUV

Summary | Events | Activities

Activities 1 to 7

Activity ID	Name	Version	Status	Schedule Time	Start Time	End Time
5	SendContractToUmzugEasyActivity.sendContractToUmzugEasy	2.0	Completed	Thursday, April 28, 2016 9:51:17 AM UTC+2	Thursday, April 28, 2016 9:51:18 AM UTC+2	Thursday, April 28, 2016 9:51:18 AM UTC+2
6	SendContractToErpActivity.sendContractOverEsbToNavision	2.0	Completed	Thursday, April 28, 2016 9:51:17 AM UTC+2	Thursday, April 28, 2016 9:51:18 AM UTC+2	Thursday, April 28, 2016 9:51:18 AM UTC+2
7	SendContractToSalesforceActivity.sendContractOverEsbToSales	2.0	Completed	Thursday, April 28, 2016 9:51:17 AM UTC+2	Thursday, April 28, 2016 9:51:18 AM UTC+2	Thursday, April 28, 2016 9:51:18 AM UTC+2
4	MapContractJsonToCdmActivity.mapJsonToCdm	2.1	Completed	Thursday, April 28, 2016 9:51:07 AM UTC+2	Thursday, April 28, 2016 9:51:07 AM UTC+2	Thursday, April 28, 2016 9:51:07 AM UTC+2
3	GetContractByCwidActivity.getContractByCwid	1.0	Completed	Thursday, April 28, 2016 9:50:56 AM UTC+2	Thursday, April 28, 2016 9:50:56 AM UTC+2	Thursday, April 28, 2016 9:50:57 AM UTC+2
2	GetUserBySsoIdActivity.getUserBySsoId	2.0	Completed	Thursday, April 28, 2016 9:50:46 AM UTC+2	Thursday, April 28, 2016 9:50:46 AM UTC+2	Thursday, April 28, 2016 9:50:46 AM UTC+2
1	GetCustomerByCwidActivity.getCustomerByCwid	2.0	Completed	Thursday, April 28, 2016 9:50:35 AM UTC+2	Thursday, April 28, 2016 9:50:35 AM UTC+2	Thursday, April 28, 2016 9:50:35 AM UTC+2

AWS CloudWatch provides performance metrics for your workflows



☐	TUV	Upsert Contract	1.0	StartedDecisionTasksTimedOutOnClose
☑	TUV	Upsert Contract	1.0	WorkflowStartToCloseTime
☐	TUV	Upsert Contract	1.0	WorkflowsCanceled
☐	TUV	Upsert Contract	1.0	WorkflowsCompleted

Title: WorkflowStartToCloseTime (Milliseconds) — Average — 1 Minute

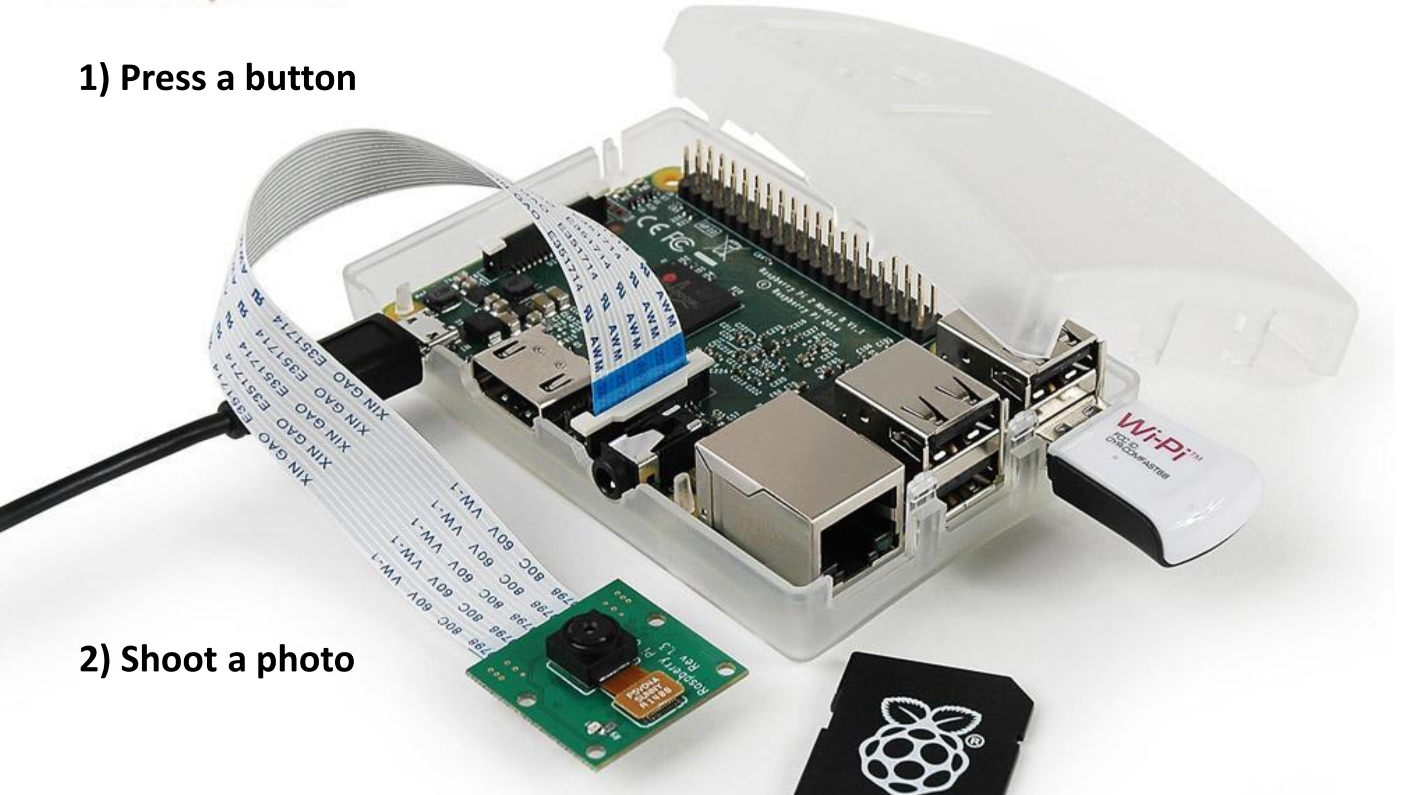
© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

- Disruptive forces and what they do with enterprise IT
- An ideal integration platform
- AWS Simple Workflows (SWF) in a nutshell
- **Demo time**
- Leveraging SWF to get rid of a classical ESB solution
- Reclaim process ownership and end-2-end-autonomy
- Drawing the big picture of a hybrid integration solution

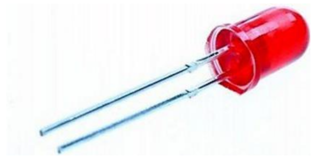
SWF-powered IoT-Photobooth

1) Press a button

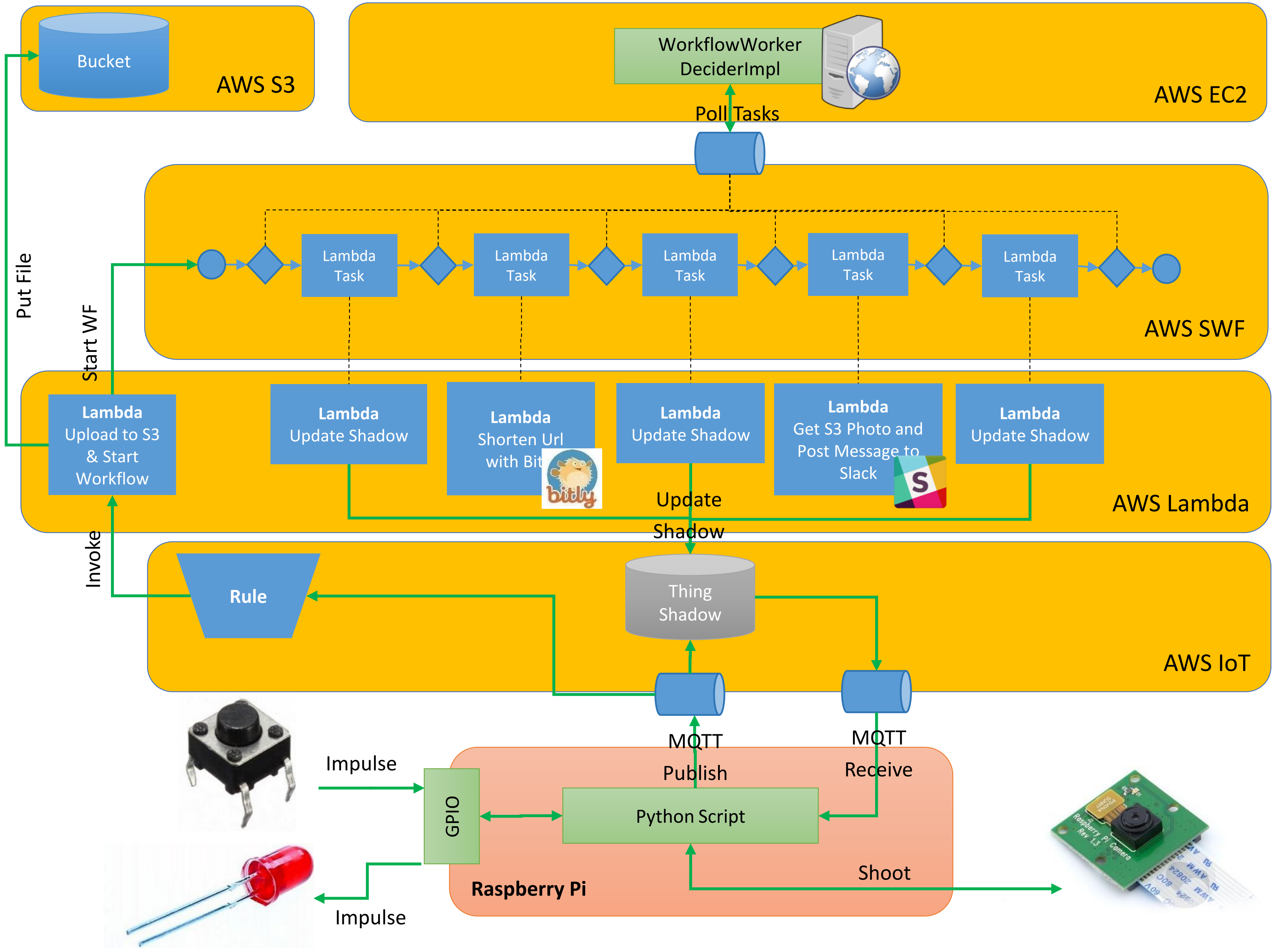


2) Shoot a photo

3) View on Slack



View progress on breadboard

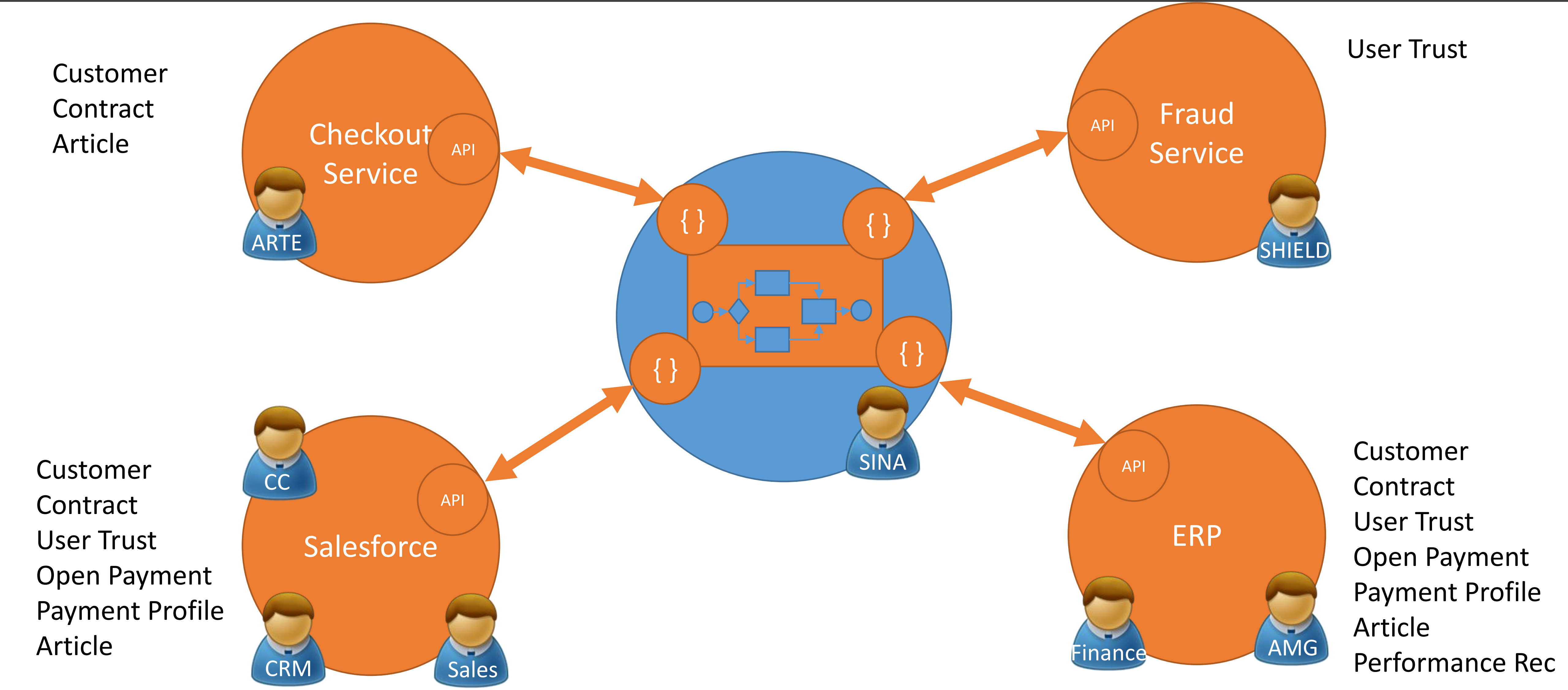


Bucket
AWS S3
WorkflowWorker
DeciderImpl
AWS EC2
Poll Tasks
Lambda Task
Lambda Task
Lambda Task
Lambda Task
Lambda Task
AWS SWF
Put File
Start WF
Lambda Upload to S3 & Start Workflow
Lambda Update Shadow
Lambda Shorten Url with Bit
Lambda Update Shadow
Lambda Get S3 Photo and Post Message to Slack
Lambda Update Shadow
Update Shadow
AWS Lambda
Invoke
Rule
Thing Shadow
AWS IoT
MQTT Publish
MQTT Receive
Impulse
GPIO
Python Script
Shoot
Raspberry Pi
Impulse

Agenda

- Disruptive forces and what they do with enterprise IT
- An ideal integration platform
- AWS Simple Workflows (SWF) in a nutshell
- Demo time
- **Leveraging SWF to get rid of a classical ESB solution**
- Reclaim process ownership and end-2-end-autonomy
- Drawing the big picture of a hybrid integration solution

So let's use SWF for replacing the ESB in our company.



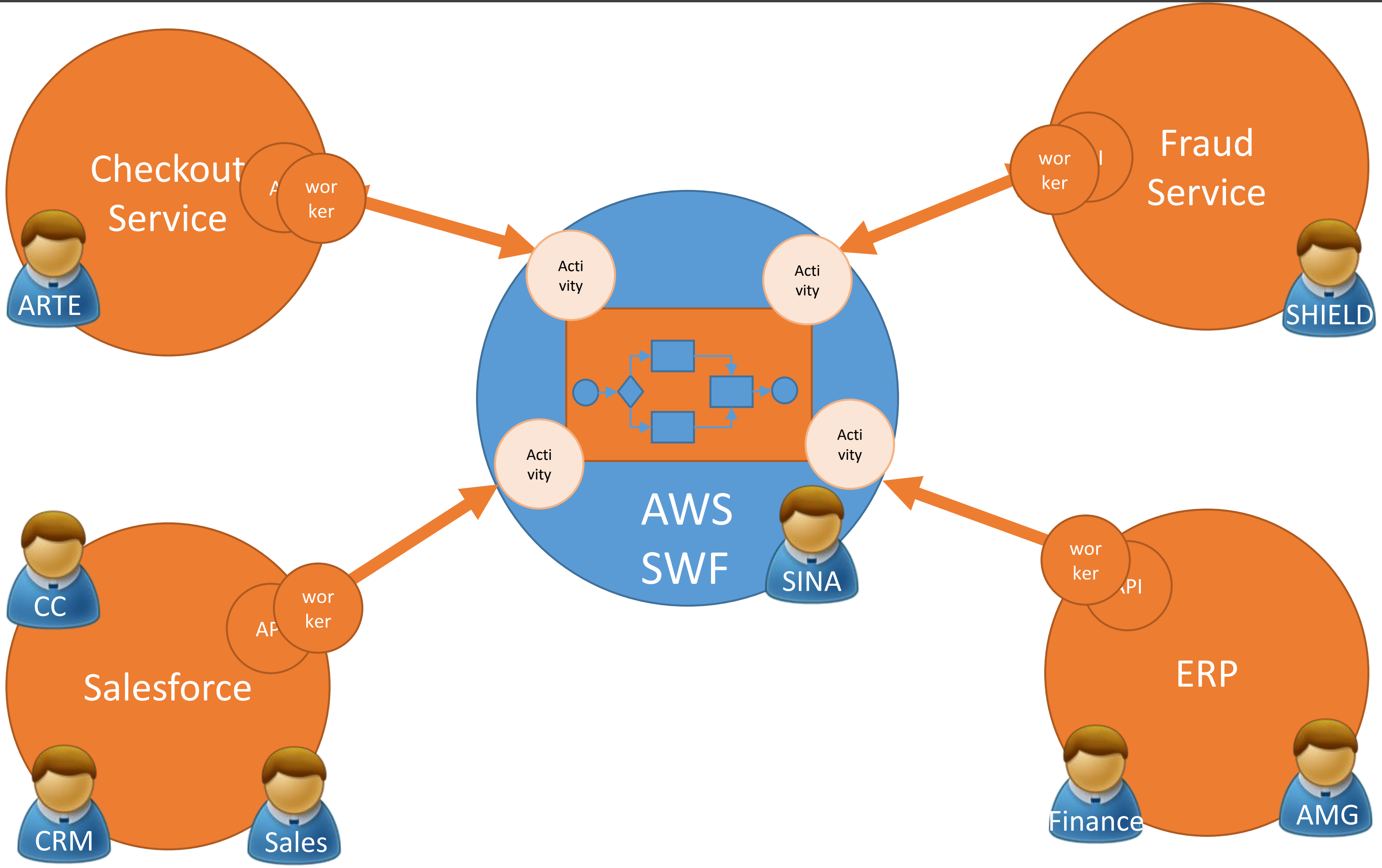
Client-side worker integrate their services by pulling tasks from SWF

Customer
Contract
Article

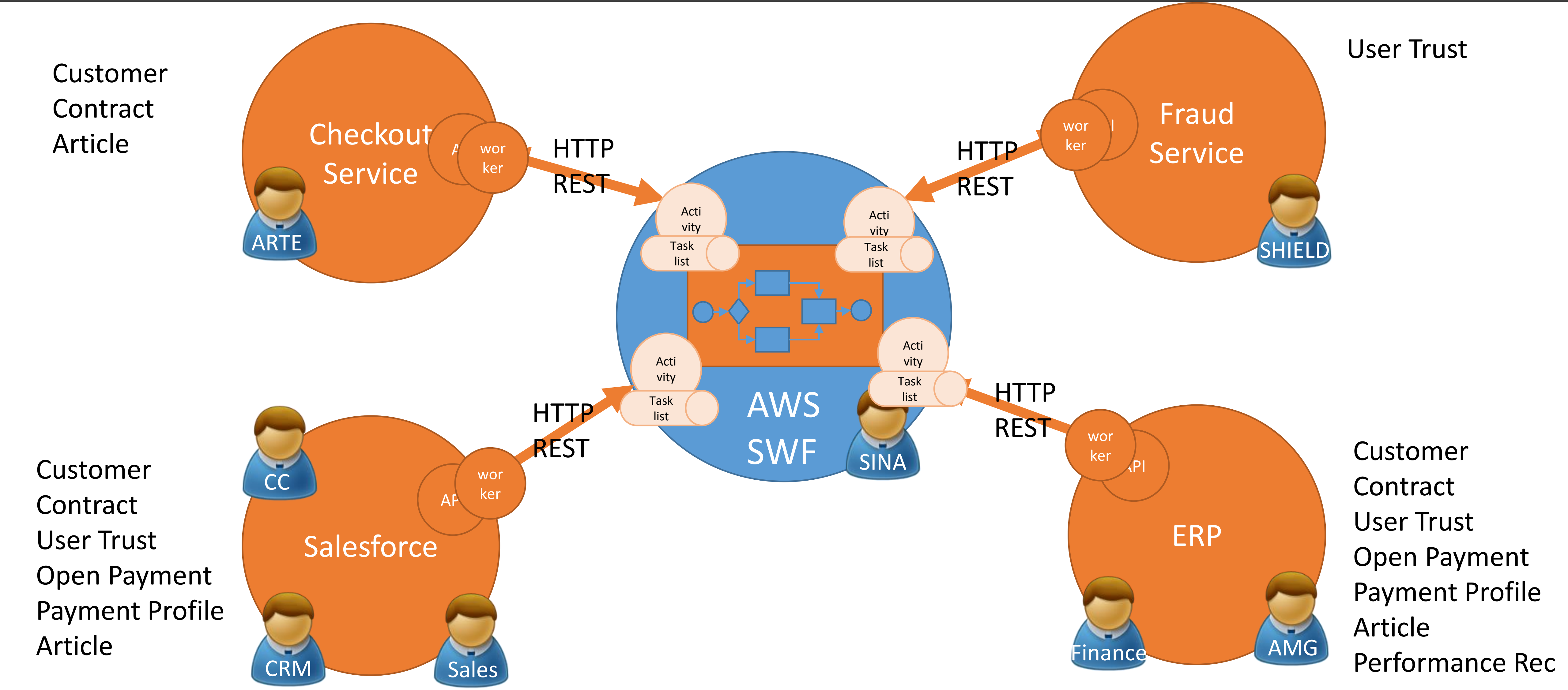
User Trust

Customer
Contract
User Trust
Open Payment
Payment Profile
Article

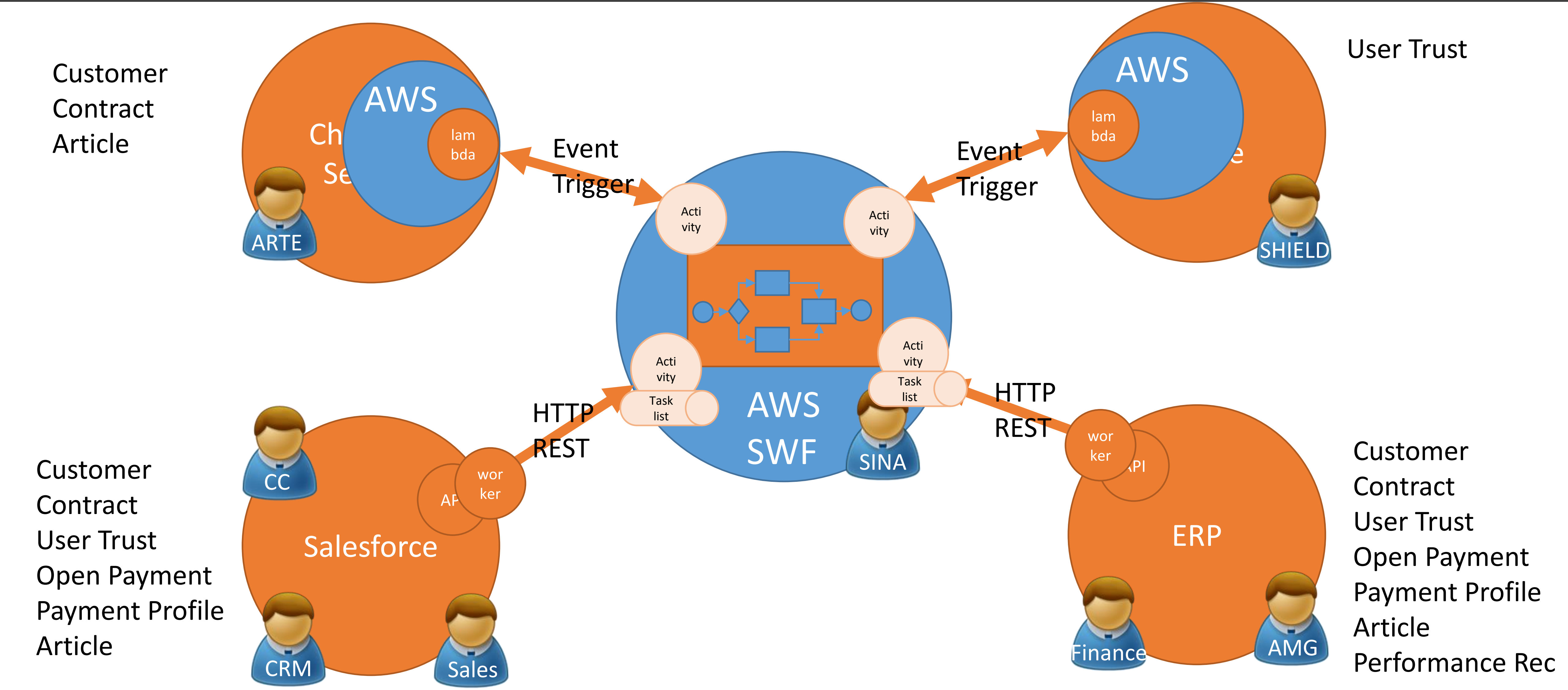
Customer
Contract
User Trust
Open Payment
Payment Profile
Article
Performance Rec



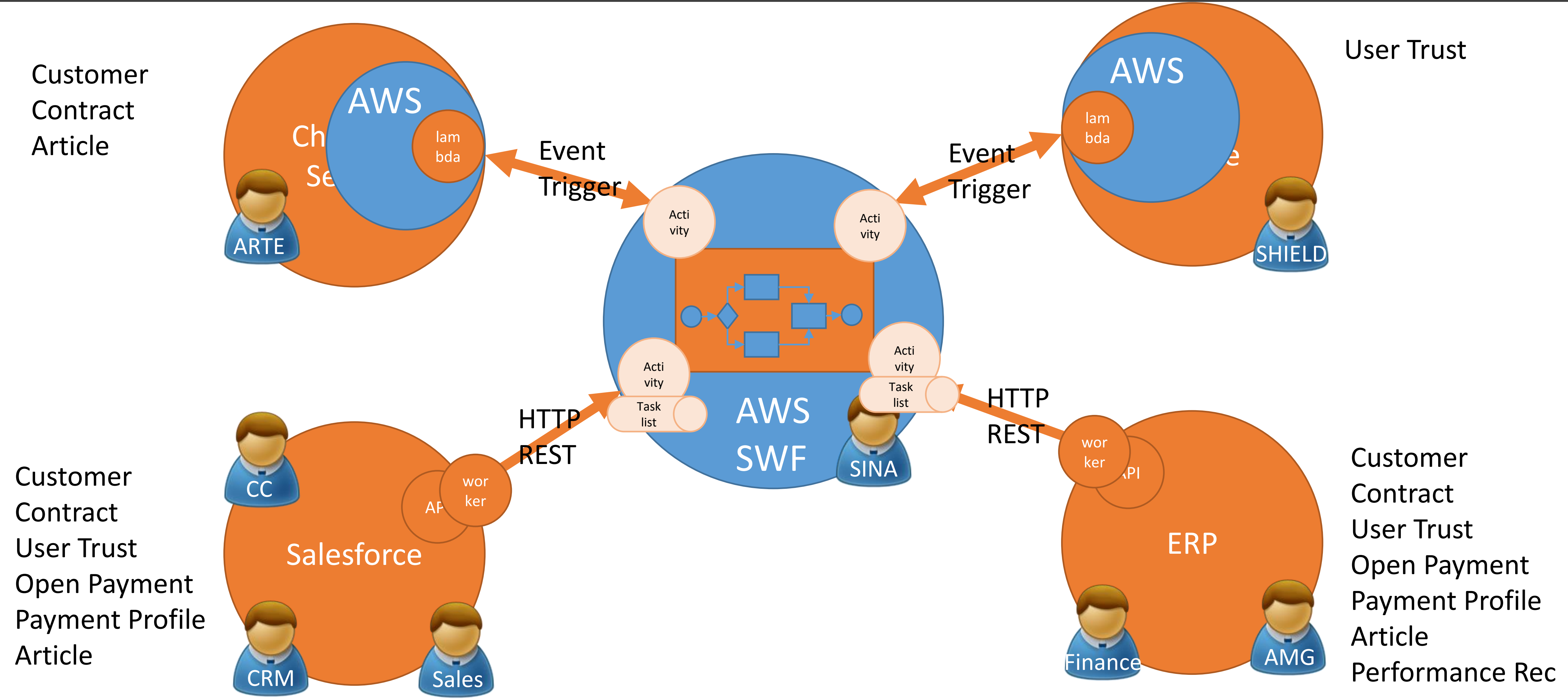
Workers poll for tasks in task lists associated with declarative activity type



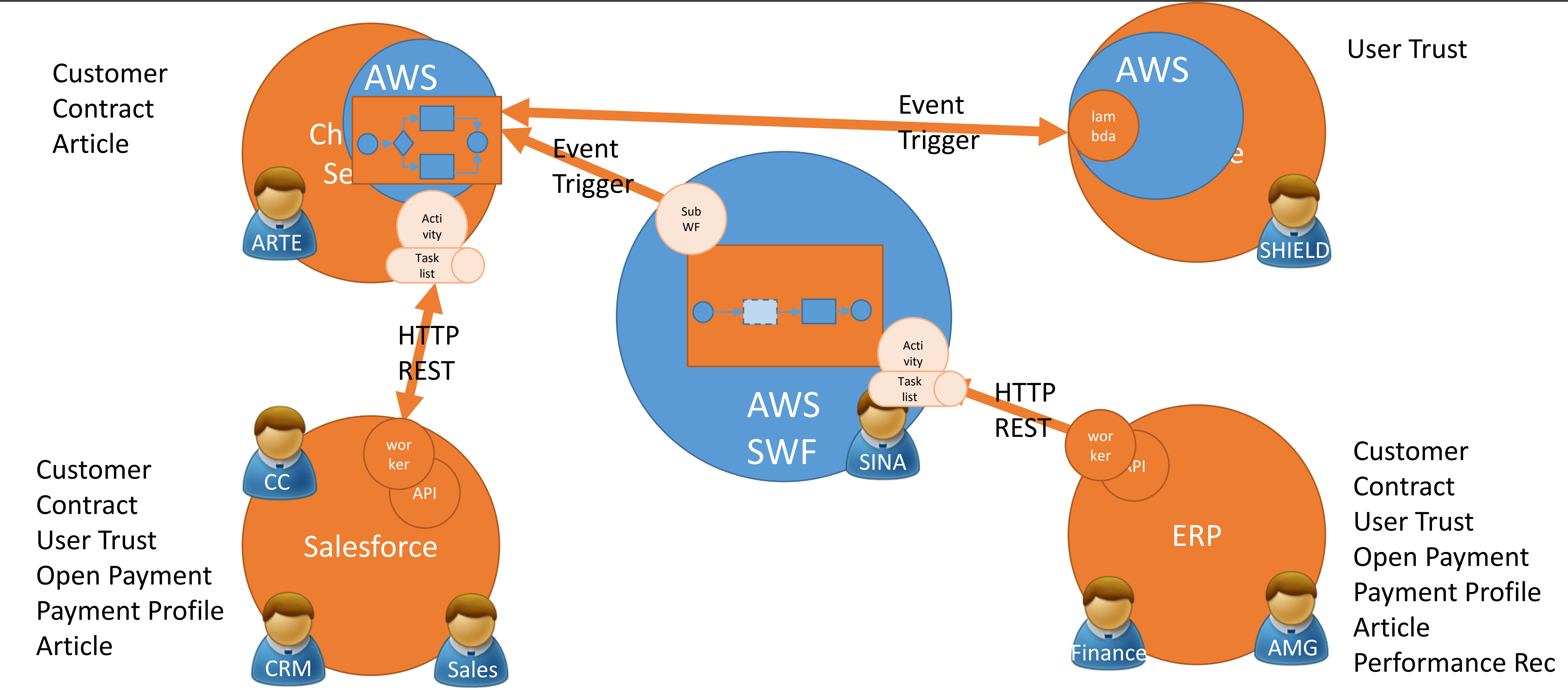
Workers can be Lambda functions as well in charge of the edge team



That’s nice, but there’s still a central orchestration controlled by one decider



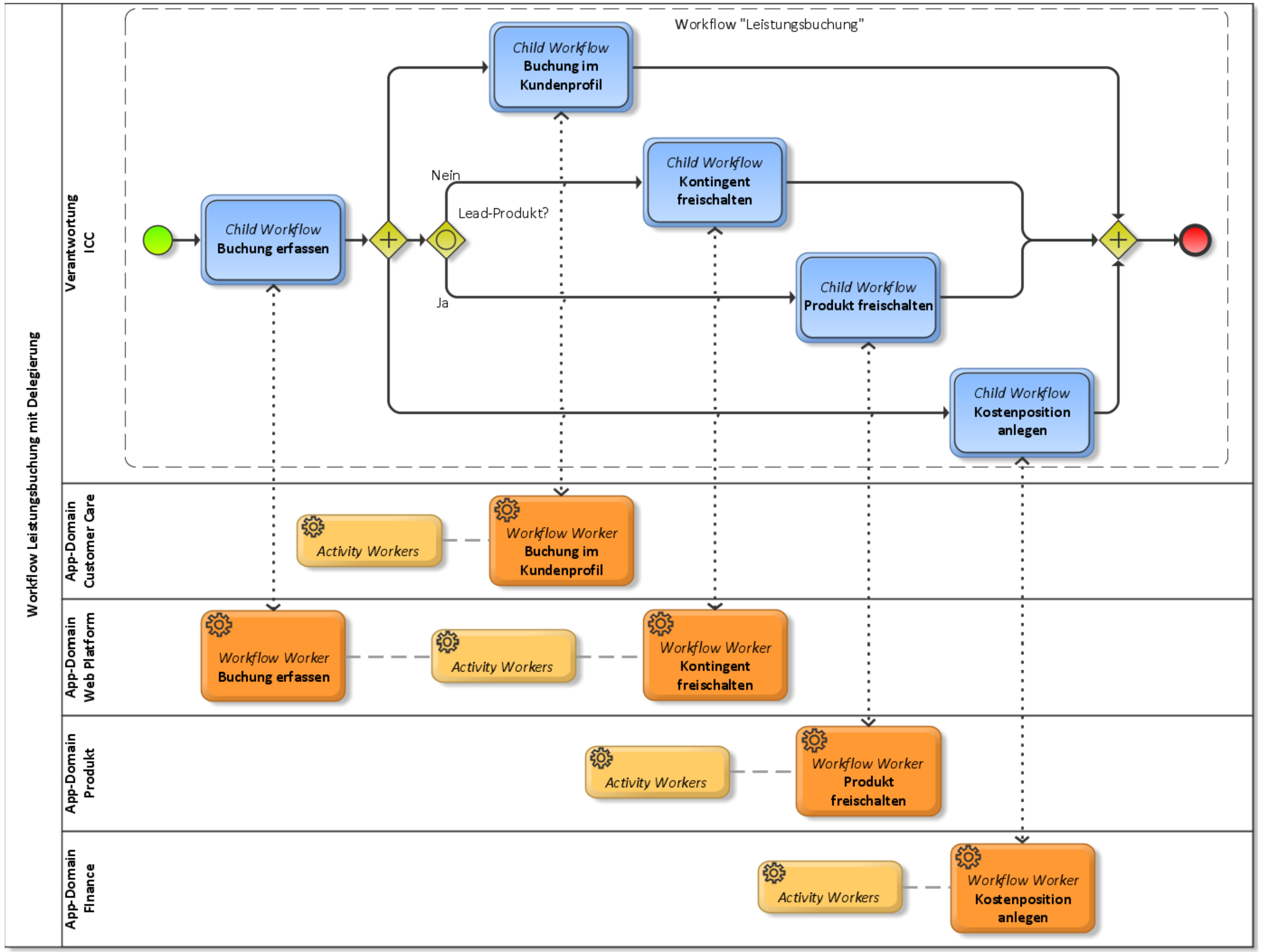
Delegate parts of the orchestration in child workflows reclaims process ownership



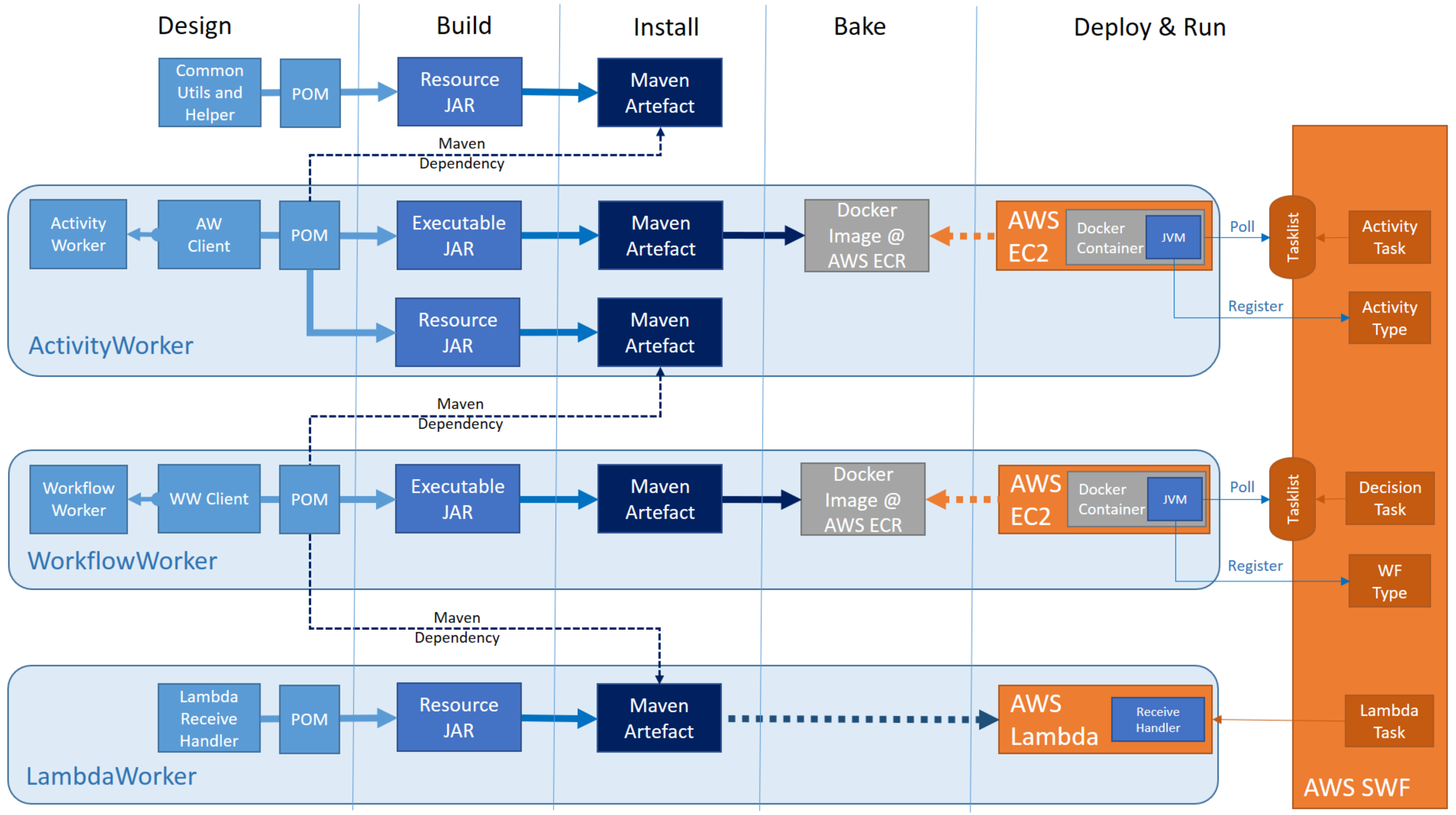
Agenda

- Disruptive forces and what they do with enterprise IT
- An ideal integration platform
- AWS Simple Workflows (SWF) in a nutshell
- Demo time
- Leveraging SWF to get rid of a classical ESB solution
- **Reclaim process ownership and end-2-end-autonomy**
- Drawing the big picture of a hybrid integration solution

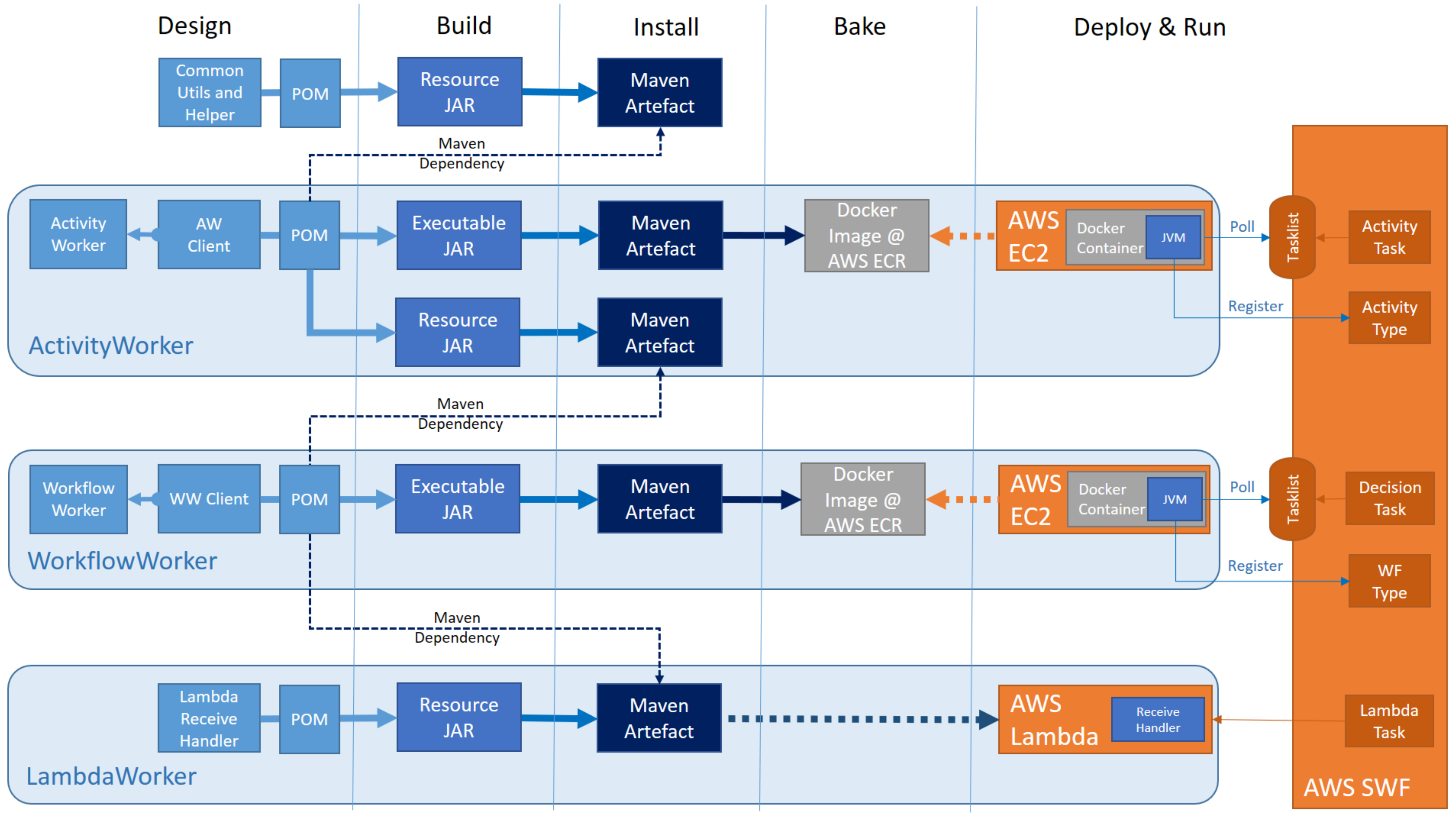
Reclaim process ownership leads to real end-2-end-autonomy of teams


Workflow Leistungsbuchung mit Delegierung
Verantwortung ICC
Workflow "Leistungsbuchung"
Child Workflow Buchung erfassen
Child Workflow Buchung im Kundenprofil
Nein
Lead-Produkt?
Ja
Child Workflow Kontingent freischalten
Child Workflow Produkt freischalten
Child Workflow Kostenposition anlegen
App-Domain Customer Care
Activity Workers
Workflow Worker Buchung im Kundenprofil
App-Domain Web Platform
Workflow Worker Buchung erfassen
Activity Workers
Workflow Worker Kontingent freischalten
App-Domain Produkt
Activity Workers
Workflow Worker Produkt freischalten
App-Domain Finance
Activity Workers
Workflow Worker Kostenposition anlegen

End-2-End autonomy along the whole lifecycle of a worker

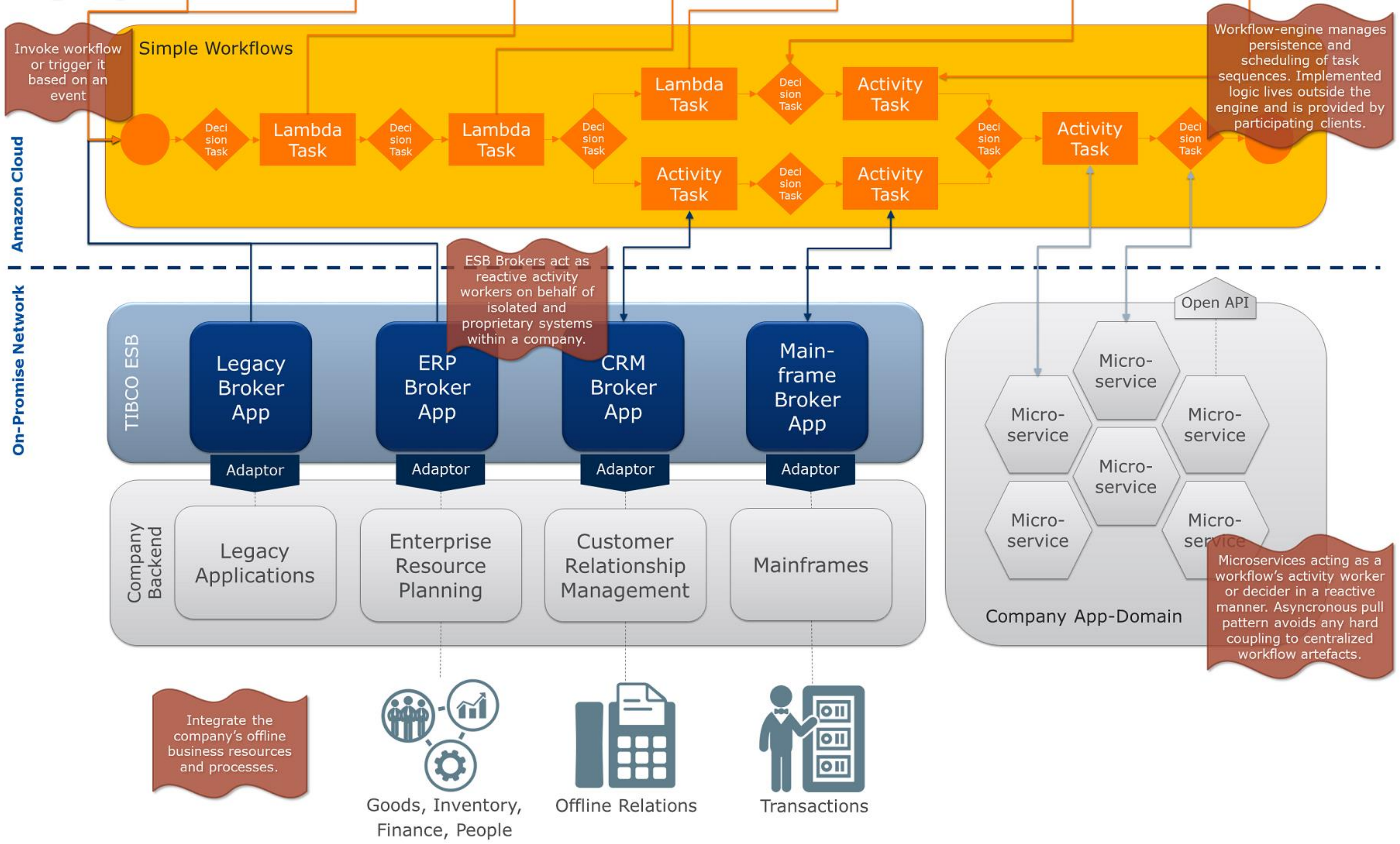


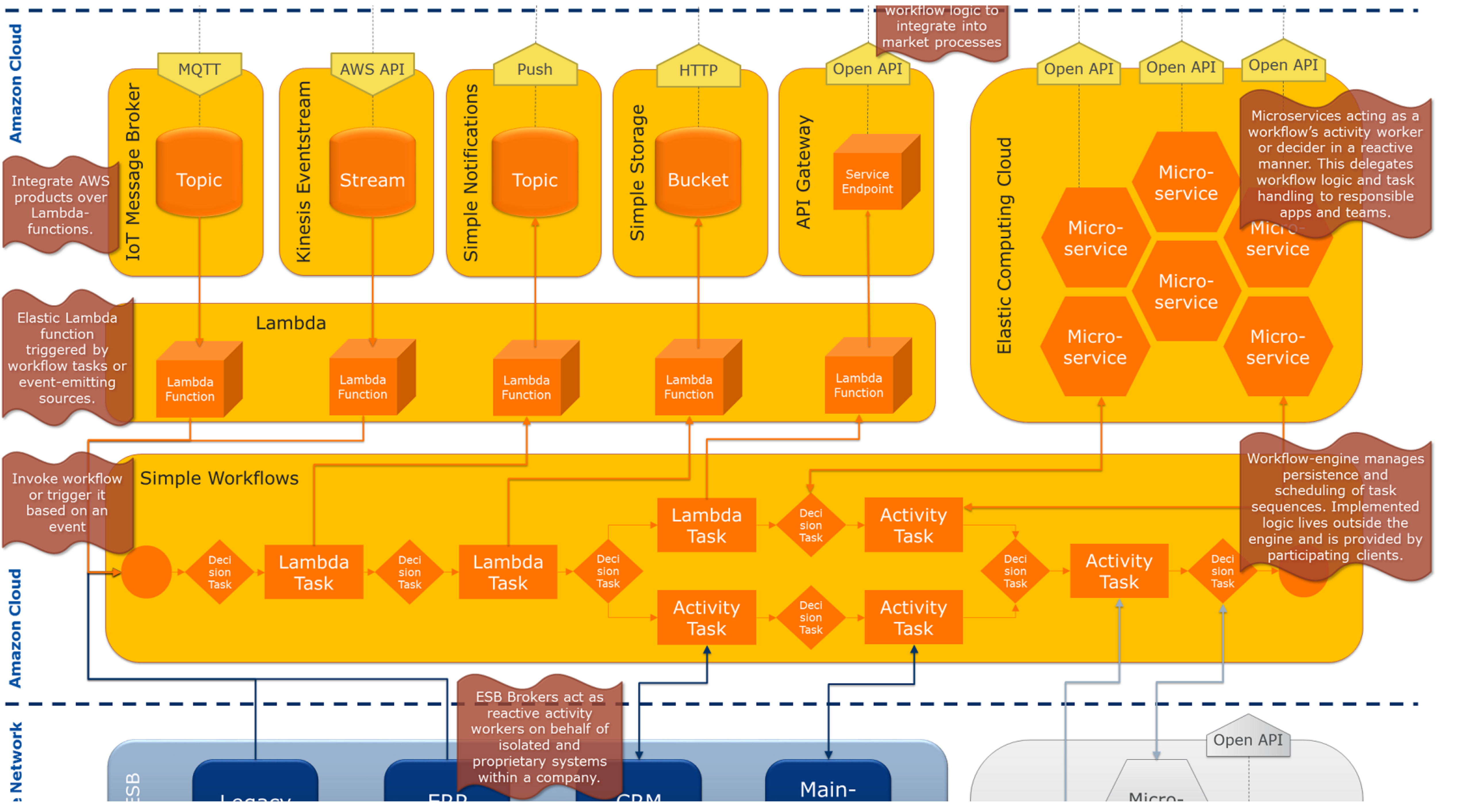
We call them Microworker as this is what they really are

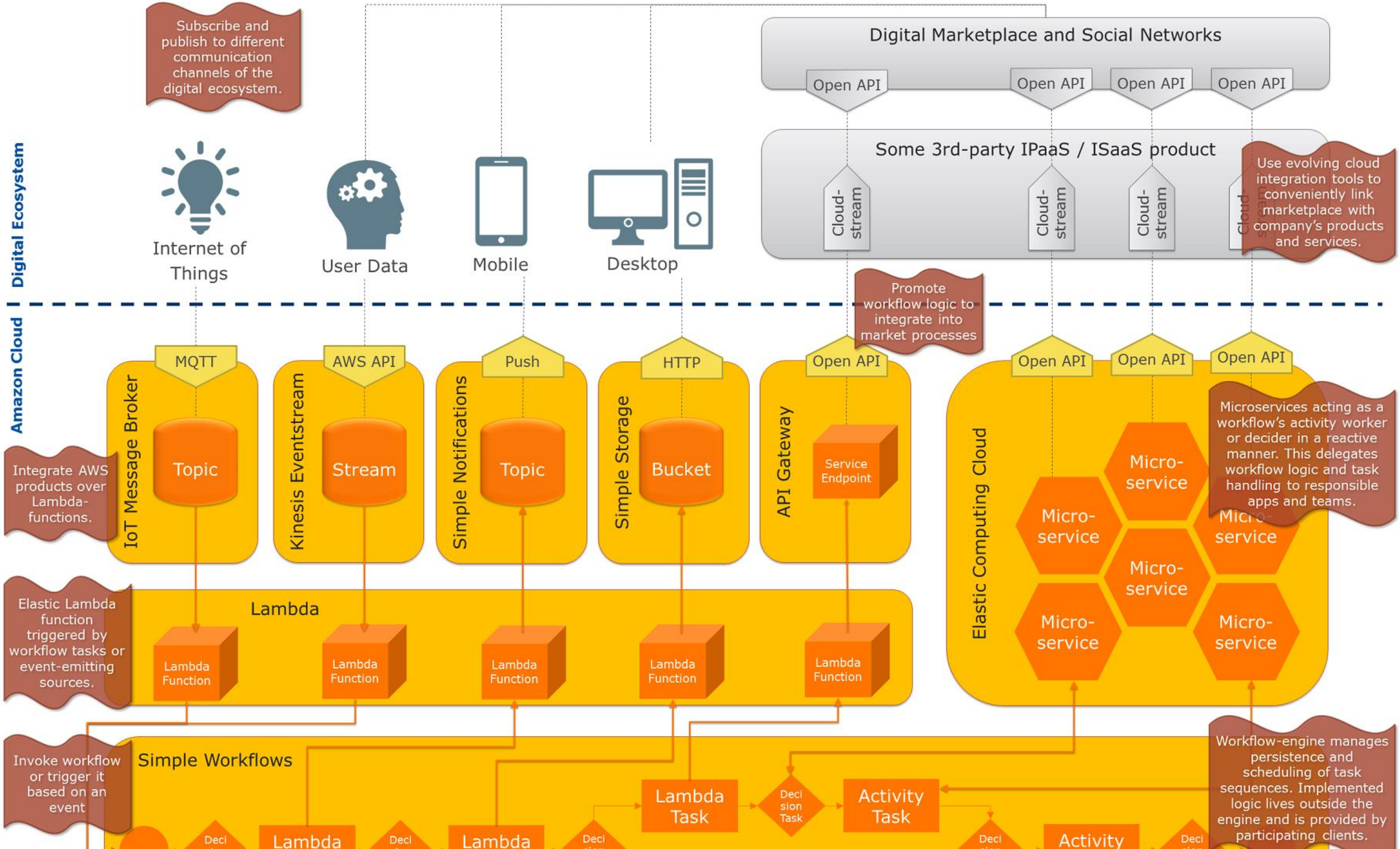


Agenda

- Disruptive forces and what they do with enterprise IT
- An ideal integration platform
- AWS Simple Workflows (SWF) in a nutshell
- Demo time
- Leveraging SWF to get rid of a classical ESB solution
- Reclaim process ownership and end-2-end-autonomy
- **Drawing the big picture of a hybrid integration solution**


Amazon Cloud
On-Promise Network
Invoke workflow or trigger it based on an event
Simple Workflows
Decision Task
Lambda Task
Decision Task
Lambda Task
Decision Task
Lambda Task
Decision Task
Activity Task
Activity Task
Decision Task
Activity Task
Decision Task
Activity Task
Decision Task
Workflow-engine manages persistence and scheduling of task sequences. Implemented logic lives outside the engine and is provided by participating clients.
ESB Brokers act as reactive activity workers on behalf of isolated and proprietary systems within a company.
TIBCO ESB
Legacy Broker App
ERP Broker App
CRM Broker App
Main-frame Broker App
Adaptor
Adaptor
Adaptor
Adaptor
Company Backend
Legacy Applications
Enterprise Resource Planning
Customer Relationship Management
Mainframes
Open API
Micro-service
Micro-service
Micro-service
Micro-service
Micro-service
Micro-service
Company App-Domain
Microservices acting as a workflow's activity worker or decider in a reactive manner. Asyncronous pull pattern avoids any hard coupling to centralized workflow artefacts.
Integrate the company's offline business resources and processes.
Goods, Inventory, Finance, People
Offline Relations
Transactions


Amazon Cloud
workflow logic to integrate into market processes
MQTT
AWS API
Push
HTTP
Open API
Open API
Open API
Open API
IoT Message Broker
Topic
Kinesis Eventstream
Stream
Simple Notifications
Topic
Simple Storage
Bucket
API Gateway
Service Endpoint
Elastic Computing Cloud
Micro-service
Micro-service
Micro-service
Micro-service
Micro-service
Micro-service
Microservices acting as a workflow's activity worker or decider in a reactive manner. This delegates workflow logic and task handling to responsible apps and teams.
Integrate AWS products over Lambda-functions.
Lambda
Lambda Function
Lambda Function
Lambda Function
Lambda Function
Lambda Function
Elastic Lambda function triggered by workflow tasks or event-emitting sources.
Simple Workflows
Invoke workflow or trigger it based on an event
Decision Task
Lambda Task
Decision Task
Lambda Task
Decision Task
Lambda Task
Decision Task
Activity Task
Activity Task
Decision Task
Activity Task
Decision Task
Activity Task
Decision Task
Workflow-engine manages persistence and scheduling of task sequences. Implemented logic lives outside the engine and is provided by participating clients.
Amazon Cloud
ESB Brokers act as reactive activity workers on behalf of isolated and proprietary systems within a company.
Network
ESB
Legacy
ERP
CRM
Main-
Open API
Micro-

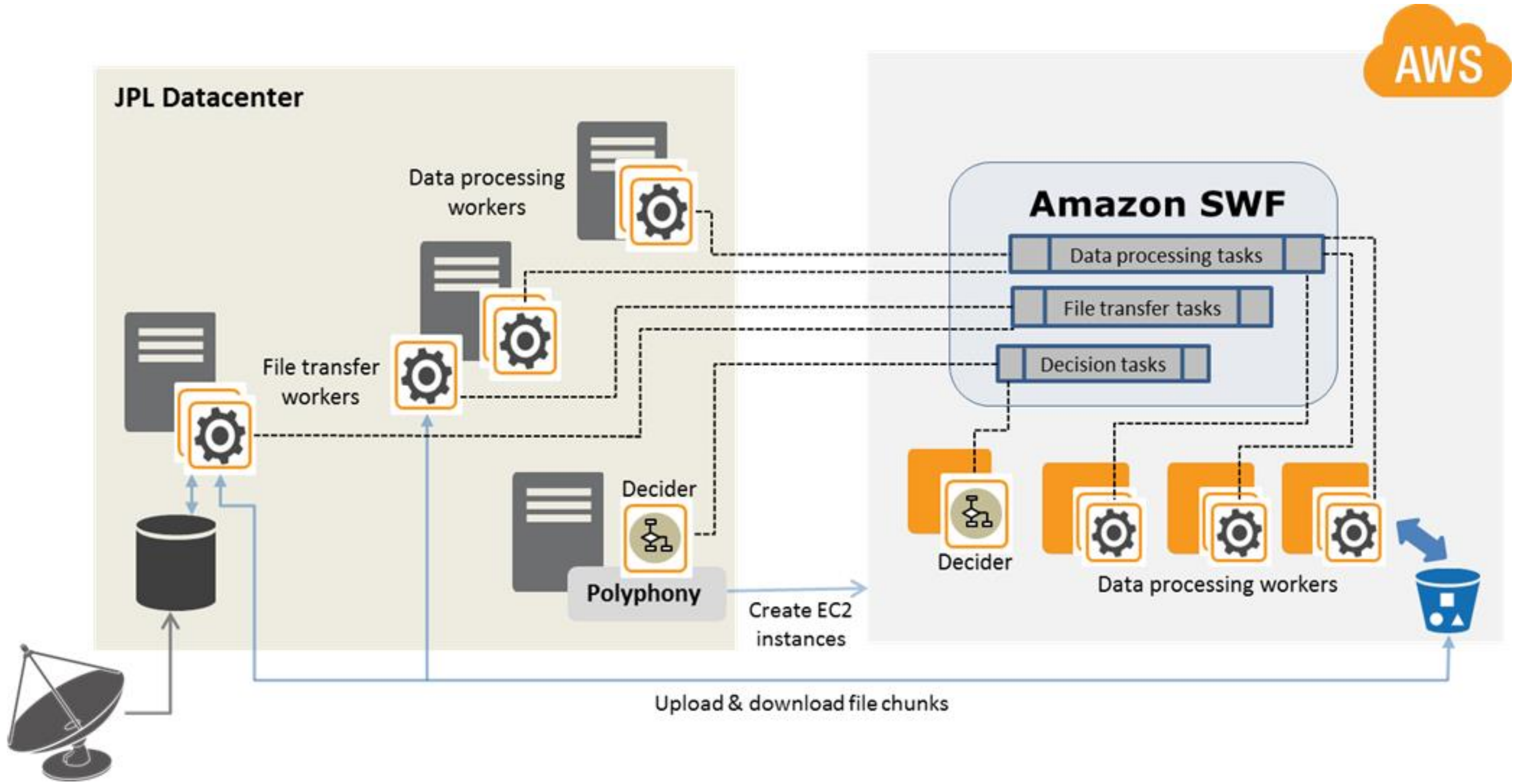

Digital Ecosystem
Amazon Cloud
Subscribe and publish to different communication channels of the digital ecosystem.
Digital Marketplace and Social Networks
Open API
Open API
Open API
Open API
Some 3rd-party IPaaS / ISaaS product
Cloud-stream
Cloud-stream
Cloud-stream
Cloud-stream
Use evolving cloud integration tools to conveniently link marketplace with company's products and services.
Internet of Things
User Data
Mobile
Desktop
Promote workflow logic to integrate into market processes
MQTT
AWS API
Push
HTTP
Open API
Open API
Open API
Open API
IoT Message Broker
Topic
Kinesis Eventstream
Stream
Simple Notifications
Topic
Simple Storage
Bucket
API Gateway
Service Endpoint
Elastic Computing Cloud
Micro-service
Micro-service
Micro-service
Micro-service
Micro-service
Micro-service
Microservices acting as a workflow's activity worker or decider in a reactive manner. This delegates workflow logic and task handling to responsible apps and teams.
Integrate AWS products over Lambda-functions.
Lambda
Lambda Function
Lambda Function
Lambda Function
Lambda Function
Lambda Function
Elastic Lambda function triggered by workflow tasks or event-emitting sources.
Invoke workflow or trigger it based on an event
Simple Workflows
Lambda Task
Decision Task
Activity Task
Workflow-engine manages persistence and scheduling of task sequences. Implemented logic lives outside the engine and is provided by participating clients.
Lambda
Lambda
Activity

Q & A

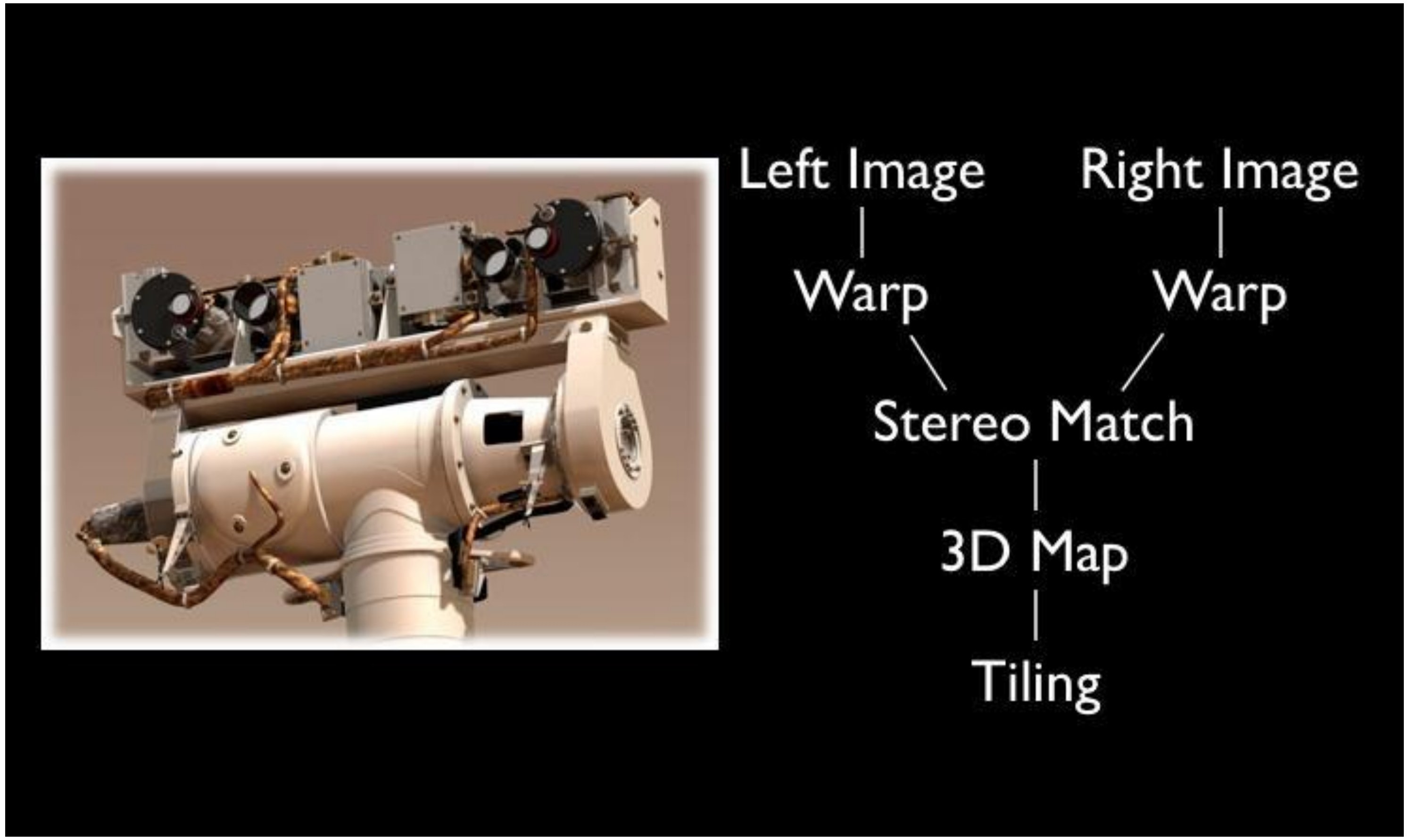
It‘s time to ask questions and give feedback.

Thanks for joining my session ...

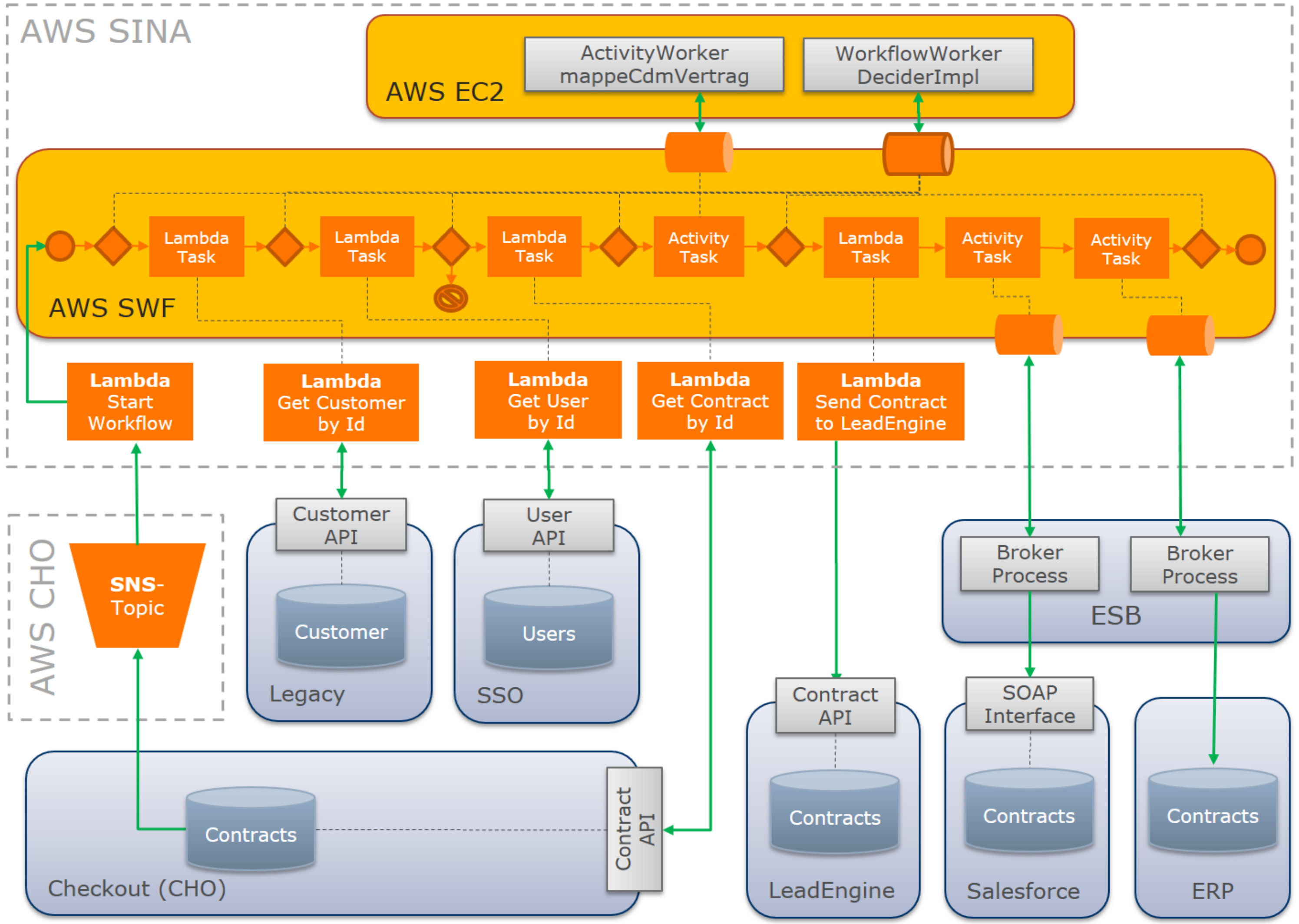
Backup (SWF at JPL Datacenter of NASA)



Backup (SWF-powered processing of images from Mars rover)



Backup (SWF-powered contract data distribution at Scout24)



Backup (SWF-powered contract data distribution at Scout24)