

March 22, 2017

Microservices 101

Overview of the Microservices Initiative at AT&T

Shridhar Rangarajan

Principal – System Architect



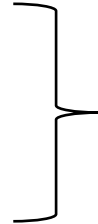
Agenda

The Business Objectives

Speed

Cost

Quality



*This discussion is about addressing these **THREE** principal business objectives by employing Microservices as the technology enablers*

What is a Microservice?

General Principles & Definition

Microservice Roadmap Strategy

Example: OMS Decomposition

Microservice Candidate Identification

Scoring Model

High-Level Plan of Action

Microservice Program Management

Program Overview




Program Setup & Resources

Q & A

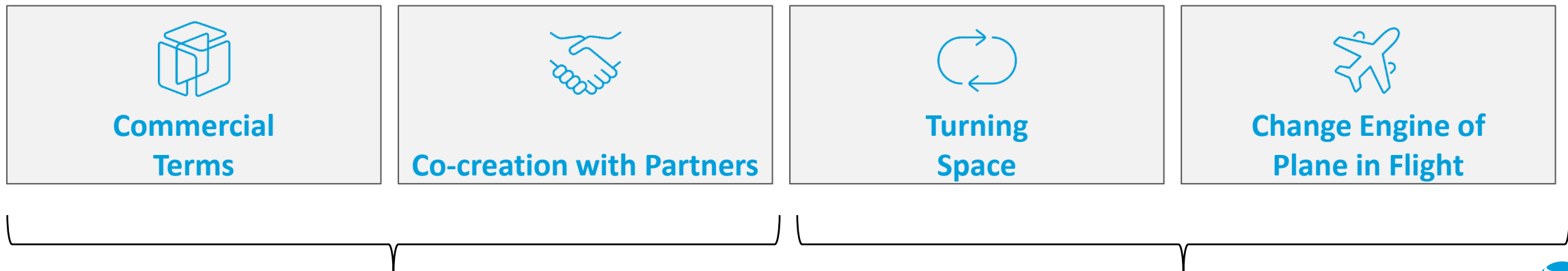
Appendix



The Business Objectives

 Speed / TTM	<ul style="list-style-type: none">– Smaller projects– Automated testing/deployment– Reduced overhead
 Cost / Reuse	<ul style="list-style-type: none">– Normalized Business Functions– Single capability focus– Accelerated revenue generation
 Quality / Availability	<ul style="list-style-type: none">– Independent– Encapsulated data– Functional isolation

4 Keys to Success



2016 Data on Application Size Impact on Speed-Cost-Quality

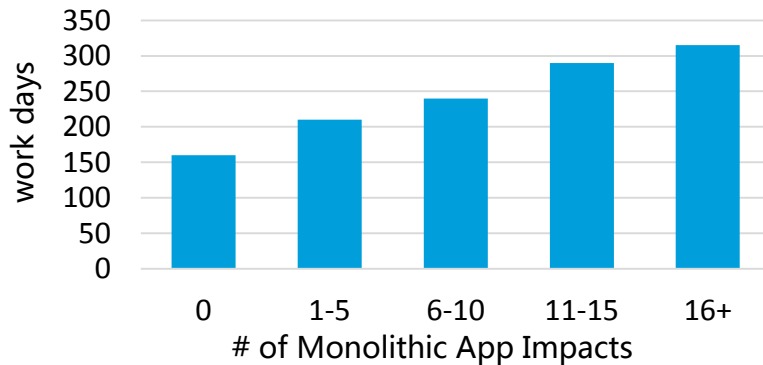
*If we define monolithic as
over 1M lines of code*

*3% of our application portfolio is
monoliths and is impacted by over 64% of
our projects*

*And the impact of a project
touching at least 1 monolith is...*

Speed

Avg. Time to Deliver by # of Monolithic App Impacts



Speed 
Monoliths

as # of



Cost

Average Cost

Non-
Monolithic
App
\$571K

Monolithic
App
\$1,876K

*Monoliths = 3-5x more
expensive*

Quality

Average # of Production Incidents

Non-
Monolithic
App
.71

Monolithic
App
3.36

*Monoliths = 5x more Prod
incidents*

















What is a Microservice?



Microservices – Definition

An architectural style in which applications are composed from loosely coupled API services with *automated & independent* lifecycles

4 Characteristics & 10 Principles

 Behavior	1 Single Capability Focused 
	2 Independence 
	3 Encapsulation 
 Control	6 Automation 
	7 Discoverable 
	8 Lightweight Communications 
 Reusability	4 Consumer First 
	5 Infrastructure Agnostic 
 Resiliency	9 Highly Resilient & Secure 
	10 Highly Observable 



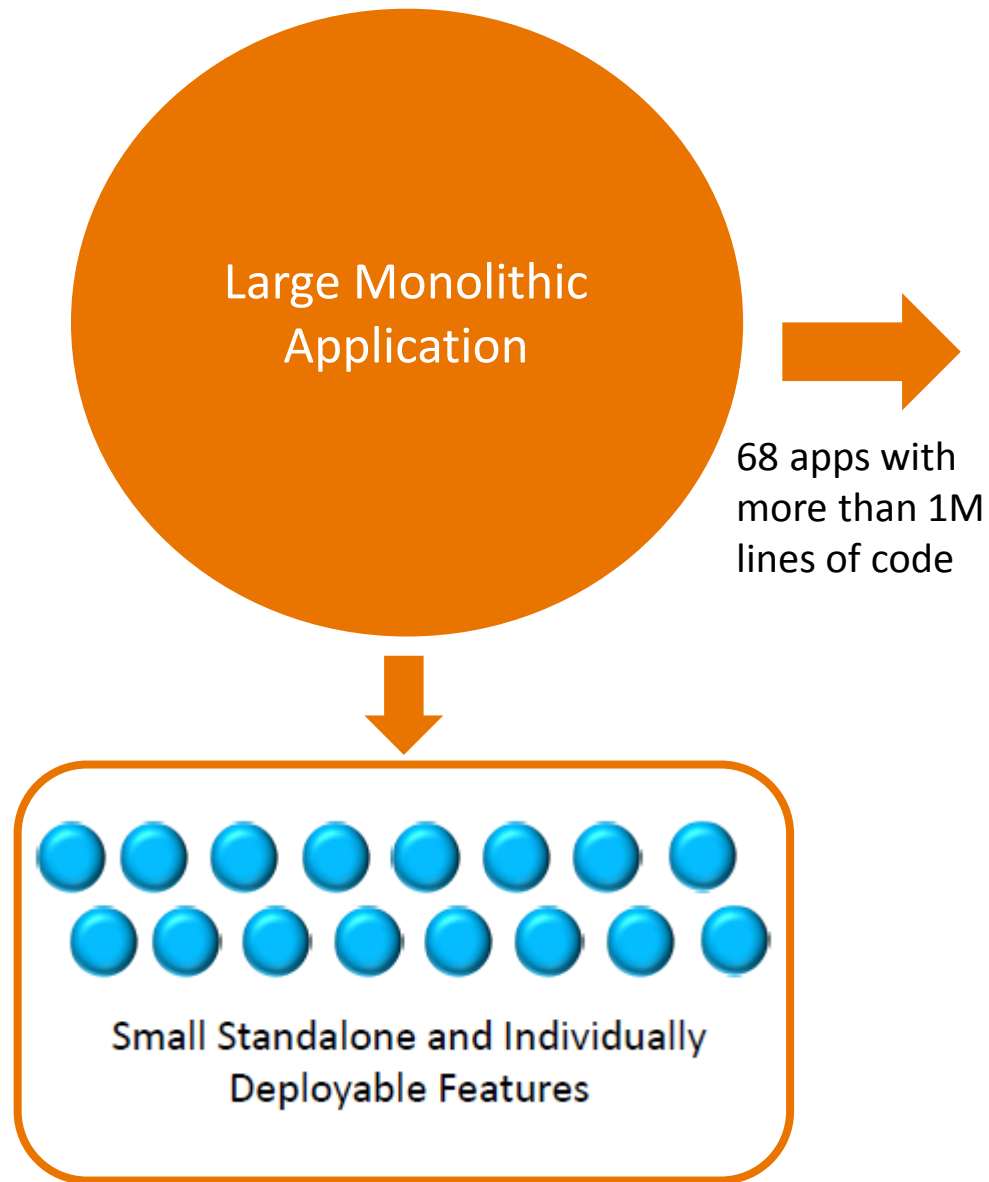
= Enabled by design practices



= Enabled by common frameworks



Breaking down Monolithic Applications using Microservices (*Application / Functional Decomposition*)



Application	Vendor	Lines of Code	Projects Impacted
SPP		23,654,900	150
RUBY		16,096,509	231
CSI		12,363,716	795
FIRST		10,404,448	135
TLG-MOB	AMDOCS	10,340,022	364
GCP-EBDP		9,499,883	2
CANOPI	AMDOCS	9,401,942	279
INSTAR		9,097,814	355
GIOM		7,933,095	260
eCDW		7,130,345	522
OMX		6,613,190	123
GCP-EIN		6,147,444	
BBNMS-LS		5,811,103	192
ORACLE SCM		5,407,365	178
Business eBill		5,243,561	182
FBF	AMDOCS	5,235,231	125
MyATT, MSe		5,017,523	328
EFMS		4,937,519	314
PRISM		4,932,746	28
OPUS - C		4,596,357	303
UB		4,461,581	203
ISAAC WFE	MOTIVE, INC.	4,461,470	169
EDGE		3,935,839	150
SDP		3,284,652	63
GRANITE	ERICSSON	3,251,965	69
Premier		2,986,539	170
CADM		2,727,756	234
WBCM		2,725,878	56
ATTeBiz		2,525,731	127
GPS		2,498,074	86
ICORE ABS		2,431,891	199
USRP		2,302,668	237
Phoenix - C		2,255,676	153
GCP-EAN		2,219,396	1

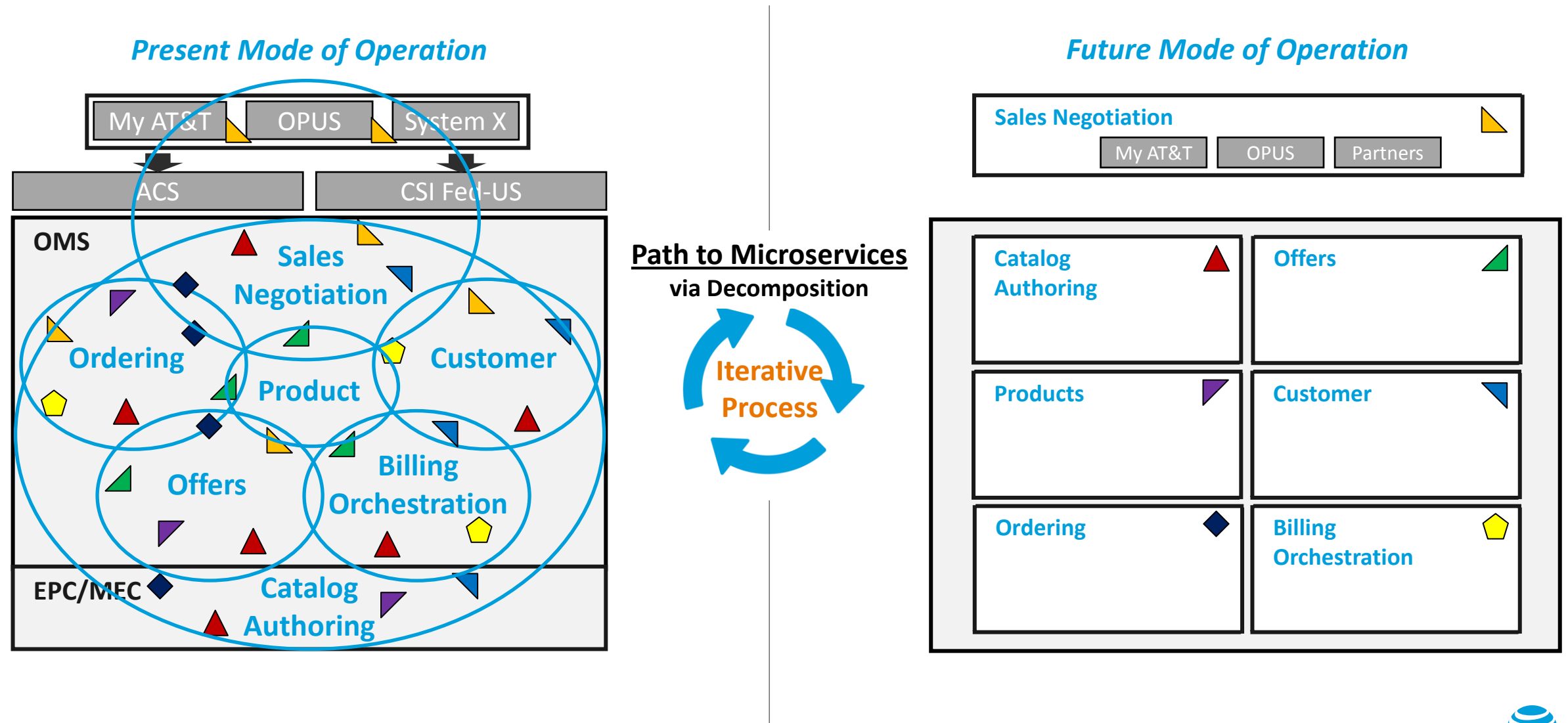
Application	Vendor	Lines of Code	Projects Impacted
PMOSS_COMPASS		2,086,615	62
BDS		2,059,179	146
PMOSS VoIP		2,058,324	107
ISB		2,021,370	24
SAART		2,007,040	98
GCAS Web		1,942,747	85
Atlas - C	TIBCO SOFTWARE, INC	1,910,763	118
myAT&T Sales		1,874,617	239
PMOSS UVerse		1,774,408	35
INSIGHT:BCTS		1,760,772	95
EDW		1,698,738	137
CS FOBPM		1,670,636	251
Enabler LS	AMDOCS	1,598,046	261
PMOSS_WIPM		1,589,969	32
WebTrax - C		1,578,010	45
CTP		1,574,717	134
ASOC		1,496,579	234
uDAS		1,422,795	74
CAPM		1,398,741	191
LS CRM	AMDOCS	1,378,615	237
ROME	ORACLE	1,363,970	419
CCDM		1,345,740	36
EM ABS		1,304,226	110
CS BOBPM		1,288,696	119
GFP-NetScope		1,279,106	62
CRE-CDMS		1,264,416	8
eTRACS	CGI GROUP, INC.	1,263,171	70
LS-OMS	AMDOCS	1,159,881	256
PMCM		1,149,102	78
CFM	SAP	1,109,575	153
Enabler - C	AMDOCS	1,096,919	77
ECBR		1,081,901	28
CCM, Clarify, CM	AMDOCS	1,073,973	119
EUAM		1,071,797	190

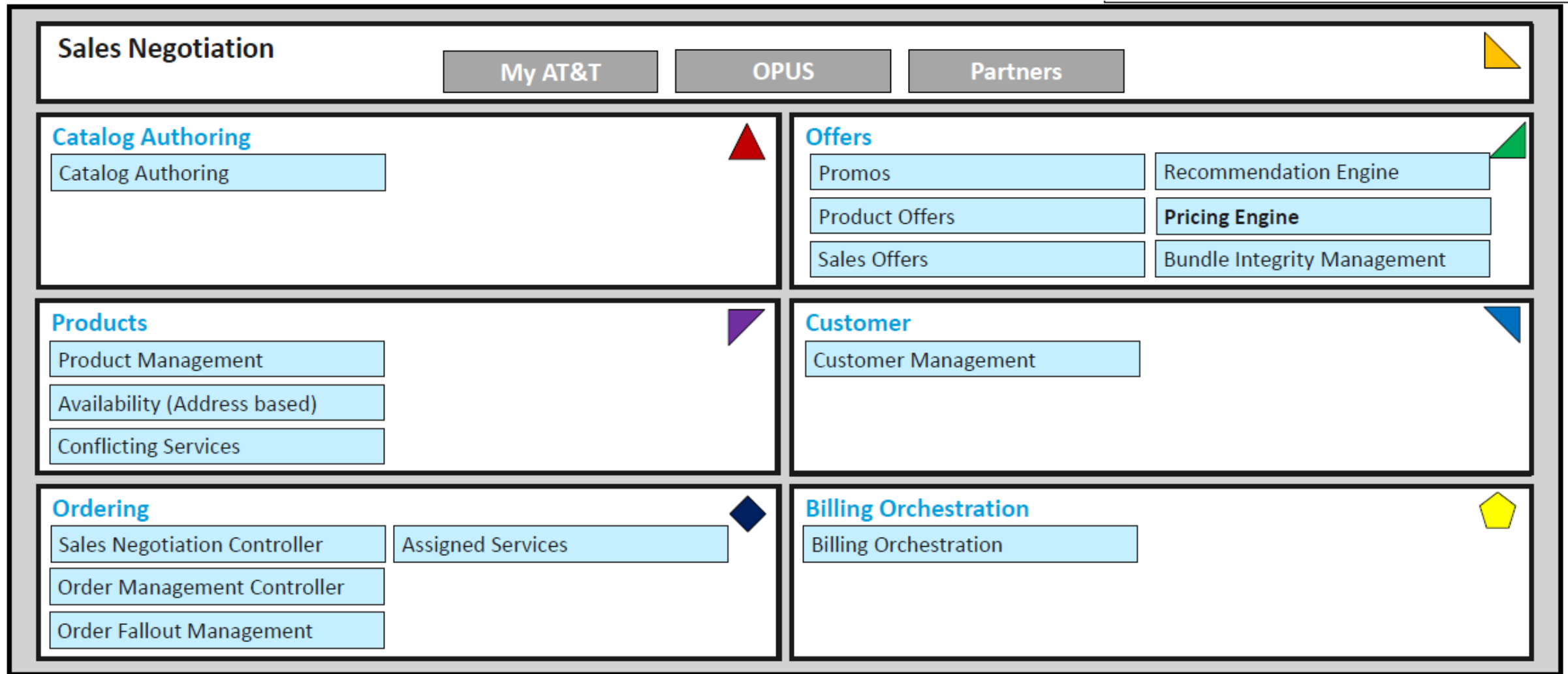
Highlighted top 10 apps with most project impacts

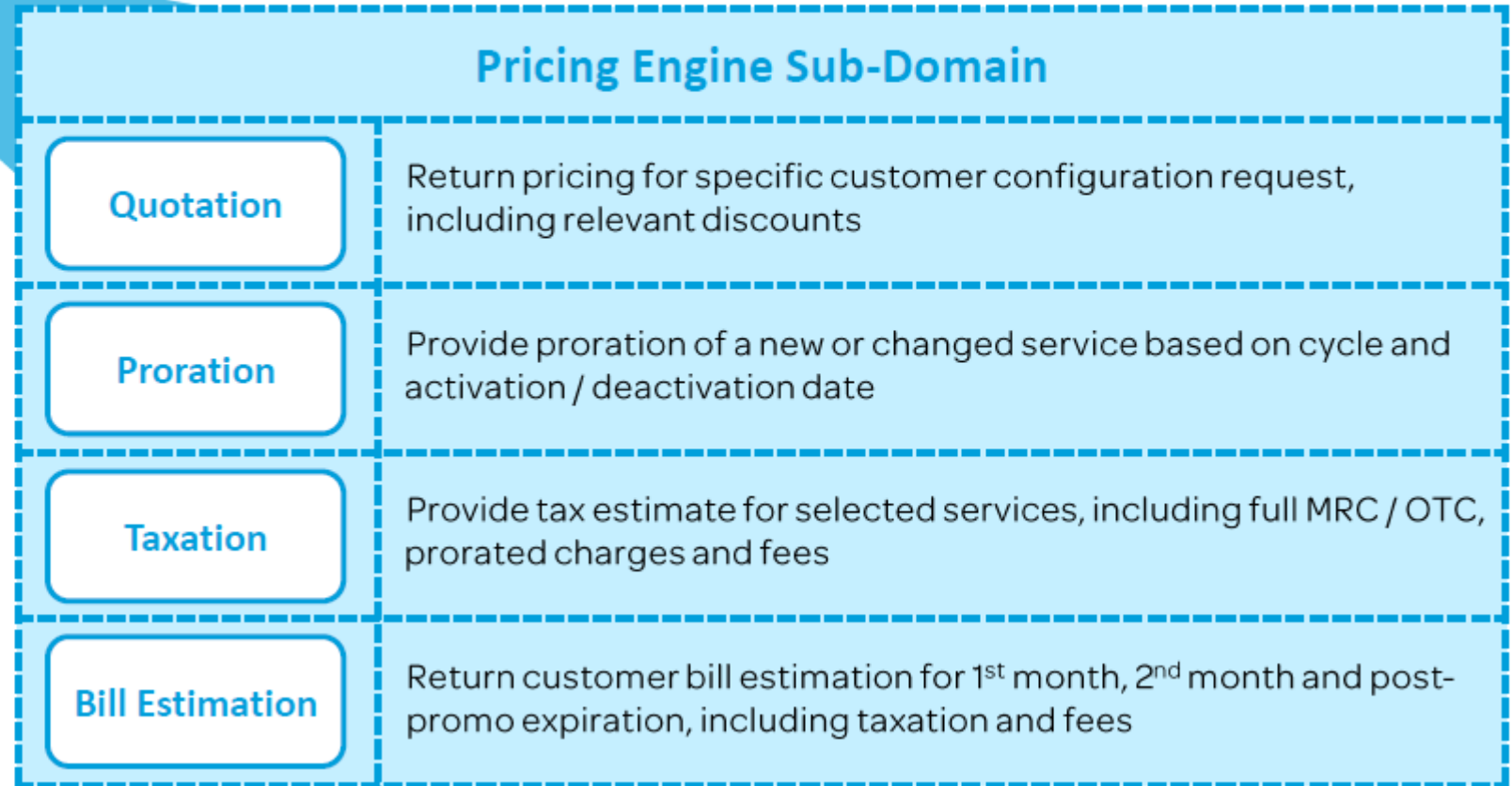
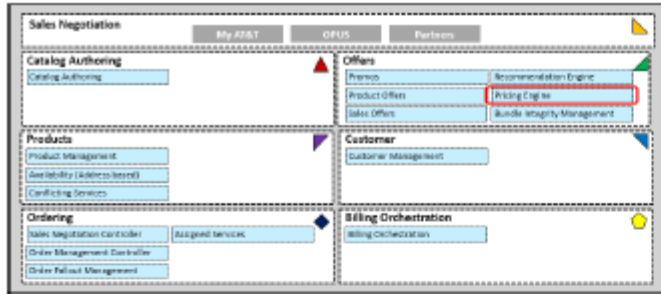
Microservice Roadmap Strategy

Example: OMS Decomposition



OMS 1st Level Decomposition (Domains)

OMS 2nd Level Decomposition (Sub-Domains)

OMS 3rd Level Decomposition (Microservices)

Microservice Candidate Identification



Business Case Framework - Bottom-up Sub-Domain Strategy by Tower

EG&M

Sub-Domain A



Sub-Domain B



Information sheets
completed by SVP leads for
each candidate sub-domain

Tower Summary

	2017	2018	2019	Total
# Sub-Domains Addressed				
Estimated # of Microservices	#	#	#	#
Total Direct Cost	\$	\$	\$	\$
BAU Funding Offset	\$	\$	\$	\$
# of projects with \$ benefit	#	#	#	#
Total \$ benefit (direct savings)	\$	\$	\$	\$
# of projects with TTM benefit	#	#	#	#
Average TTM Reduction	days	days	days	
Avg. Cost/mS	\$			
Payback Period	Months			
IRR	%			

ABS

Sub-Domain A



Sub-Domain B



Information sheets
completed by SVP leads for
each candidate sub-domain

Tower Summary

	2017	2018	2019	Total
# of Sub-Domains Addressed				
Estimated # of Microservices	#	#	#	#
Total Direct Cost	\$	\$	\$	\$
BAU Funding Offset	\$	\$	\$	\$
# of projects with \$ benefit	#	#	#	#
Total \$ benefit (direct savings)	\$	\$	\$	\$
# of projects with TTM benefit	#	#	#	#
Average TTM Reduction	days	days	days	
Avg. Cost/mS	\$			
Payback Period	Months			
IRR	%			

SD&E

Sub-Domain A



Sub-Domain B



Information sheets
completed by SVP leads for
each candidate sub-domain

Tower Summary

	2017	2018	2019	Total
# of Sub-Domains Addressed				
Estimated# of Microservices	#	#	#	#
Total Direct Cost	\$	\$	\$	\$
BAU Funding Offset	\$	\$	\$	\$
# of projects with \$ benefit	#	#	#	#
Total \$ benefit (direct savings)	\$	\$	\$	\$
# of projects with TTM benefit	#	#	#	#
Average TTM Reduction	days	days	days	
Avg. Cost/mS	\$			
Payback Period	Months			
IRR	%			



Microservices – Scorecard Model (1)

1 Costs (HOM)*

Microservice Development HOM \$M Estimate	Cost estimate to develop the Microservice
Integration HOM \$M Estimate	Cost estimate to integrate Applications with the Microservice to support product(s) in initial release
Total HOM \$M Estimate	Total costs to develop and use Microservice in its initial release

2 Cost Avoidance Calculation

Current Architecture	<ul style="list-style-type: none"> Solution Application Count Average Annual Modifications Average MDE per Release Update Total Annual Spend Projection
Microservice Architecture	
Cost Avoidance	Annual \$ Savings when comparing Microservices Architecture Costs to Current Architecture Costs (Current Arch. \$ - mS Arch. \$)
ROI	The # of Years that it will take for the annual cost avoidance to pay for the costs of the Microservices Investment (mS Arch. \$/Cost Avoidance \$) + Impl. Time

3 Business Value Score [$\sum (Weight * BV_i)$]

BV1 (1)	Digital Experience Enabler?	Y=1, N=0
BV2 (1)	Supports Multiple Products?	Y=1, N=0
BV3 (1)	# of Products Impacted with Initial Release	# of Products Supported by Proposed Microservice
BV4 (3)	TTM (Time-to-Market)	<ul style="list-style-type: none"> 10->8mos. = 1 10->6mos. = 2 10->4mos. = 3
BV5 (3)	ROI	<ul style="list-style-type: none"> >2 yr. = 1 1-2 yr. = 2 <1 yr. = 3



Microservices – Scorecard Model (2)

4

Technical Ease Score [$\sum (\text{Weight} * TE_i)$]

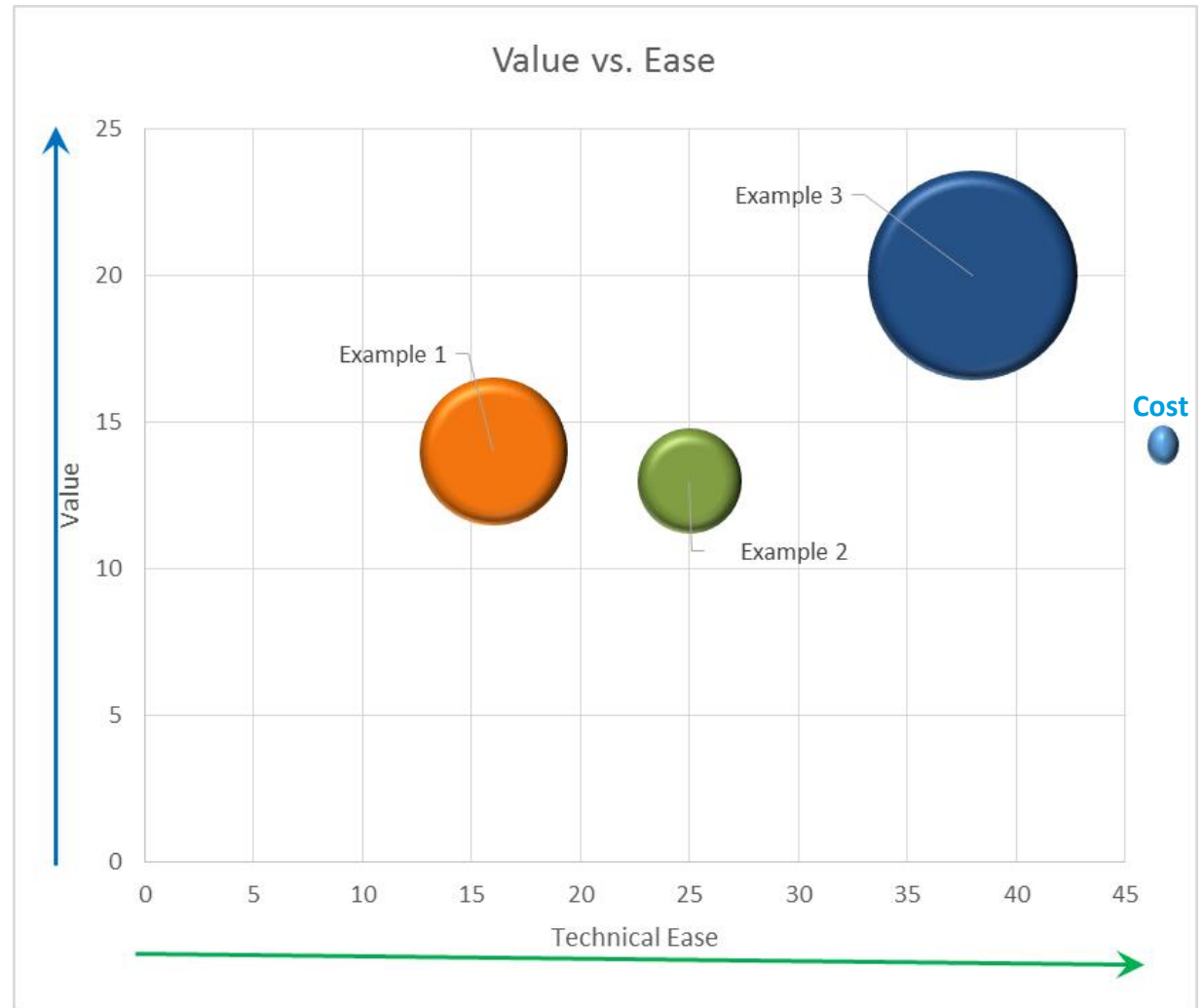
TE1 (1)	In-Year Delivery?	Y=1, N=0
TE2 (1)	Minimize Throwaway	<ul style="list-style-type: none"> • <50% = 1 • 51-75% = 2 • >75% = 3
TE3 (3)	Complexity	<ul style="list-style-type: none"> • High = 1 • Medium = 2 • Low = 3
TE4 (2)	Simplified Integration	<ul style="list-style-type: none"> • <50% = 1 • 51-75% = 2 • >75% = 3
TE5 (2)	Platform Onboarding Ease (Current Implementation in Java/J2EE?)	Y=1, N=0

TE6 (1)	Alignment with Target Architecture Roadmap	<ul style="list-style-type: none"> • High = 1 • Medium = 2 • Low = 3
TE7 (3)	Risk	<ul style="list-style-type: none"> • High = 1 • Medium = 2 • Low = 3
TE8 (1)	Dependency	<ul style="list-style-type: none"> • High = 1 • Medium = 2 • Low = 3
TE9 (2)	Data Consolidation / Reduction	<ul style="list-style-type: none"> • <50% = 1 • 51-75% = 2 • >75% = 3
TE10 (1)	Leverage Existing Data Model?	<ul style="list-style-type: none"> • <50% = 1 • 51-75% = 2 • >75% = 3



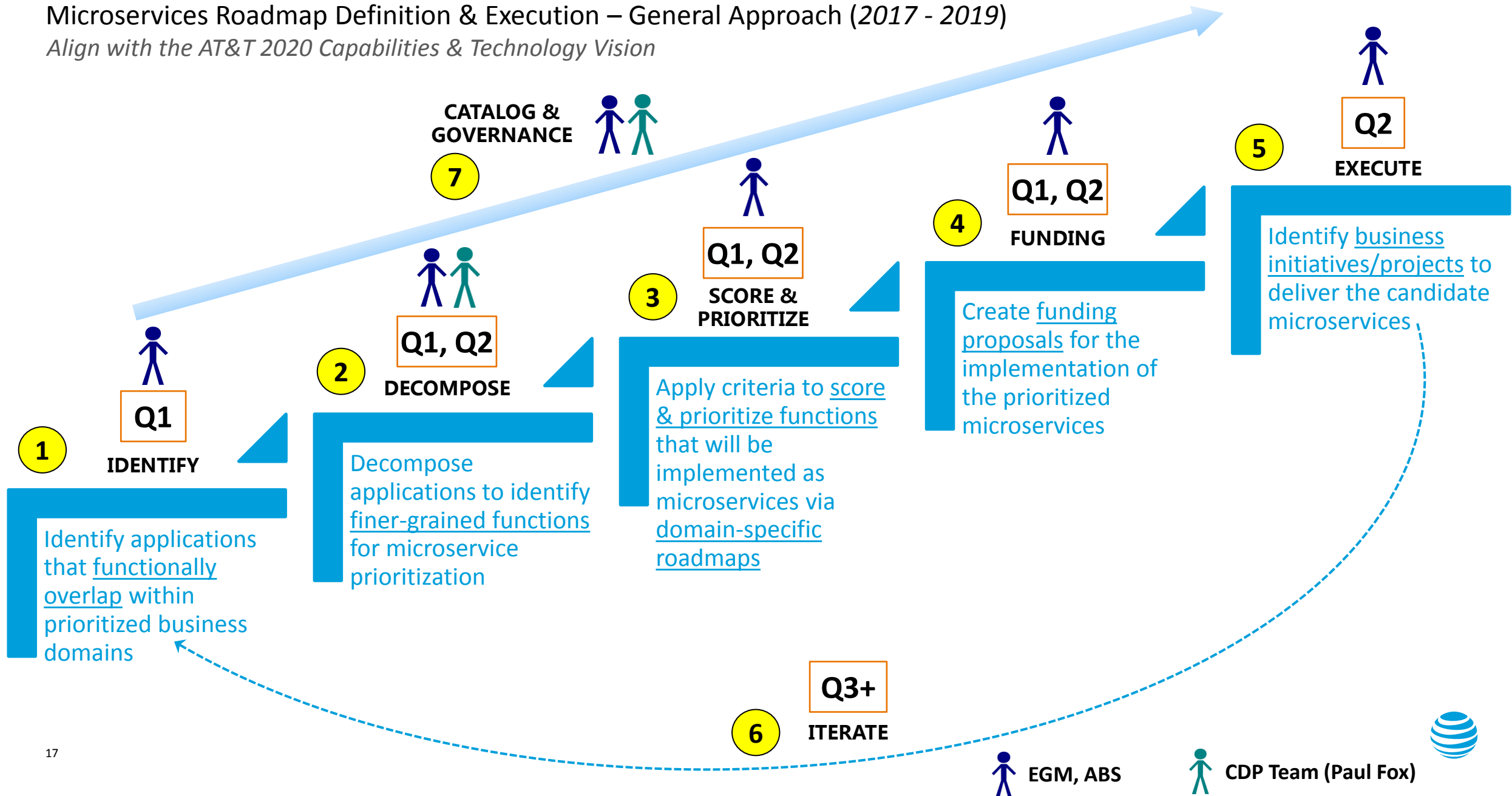
Microservices – Scorecard Chart

Microservice Domain Area	Business Value	Technical Ease	Total Costs (\$M)
Example 1	14	16	\$ 2
Example 2	13	25	\$ 1
Example 3	20	38	\$ 4



Microservices Roadmap Definition & Execution – General Approach (2017 - 2019)





Align with the AT&T 2020 Capabilities & Technology Vision



Microservice Program Management



Program Overview

 Program Management	<ul style="list-style-type: none">– Jo Anne Pate - Overall Lead– Kim Cullum– Uma Govindaraju / Celia Bengal	 <h3>2017 Progress To Date</h3> <ul style="list-style-type: none">✓ Microservices Program Established✓ Educational Webcasts Weekly✓ Microservices Candidate Scoring Model Finalized✓ Initial Candidate List Identified✓ Initial Funding Allocated✓ Microservices Catalog Established✓ Microservices Initial Business Case Developed & Presented to Leadership
 Parisian Tower	<ul style="list-style-type: none">– Lynn Morgan – Overall Lead– Rob Kafka– Henry Kozik– Joy Carter– Shridhar Rangarajan– Indu Dugan	
 Technical Standards & Architecture	<ul style="list-style-type: none">– Paul Fox – Overall Lead– Renee Glover– Mark Wyatt	









Microservices Program Setup

Track	Tower	Owner
Technical Standards & Architecture	Sorabh Saxena	Paul Fox
Software Development & Engineering	Sorabh Saxena	Monica Browning Mitchell, Dawn LaComb
Solution Architecture & MDE tProcess	Sorabh Saxena	Adam Hersch
D2 eComp	Christopher Rice	Pat Henderson
Automated Testing Strategy & Delivery	Sorabh Saxena	Mark Cottrell
Metrics	Sorabh Saxena	Olympia Connolly
Security / Resiliency	Sorabh Saxena	Hussein Fareed
ABS Prioritization & Delivery	Jon Summers	Darin Morrow / Kathy Fox
EG Prioritization & Delivery	Pam Parisian	Lynn Morgan
DE Prioritization & Delivery	Teresa Ostapower / Pam Parisian	Henry Kozik
MS DevOps & Production Support	Sorabh Saxena	Jim Morse
Data Supply Chain	Sorabh Saxena	Reem Fareed
Data Federation	Sorabh Saxena	Sapna Jain (Satish Purushothaman)
BI/BD - overall Indigo Program	Victor Nilson	Robert Gage
Sourcing & Contracting	Jenifer Robertson	Greg Snooks
Program Management	Sorabh Saxena	JoAnne Pate (Uma Govindaraju / Celia Bengel)
Program PIDs	Sorabh Saxena	Jan Harris
Training	Sorabh Saxena	Satish Purushothaman



Program Resources

Want to know more? See the helpful links below.

 Join the tSpace Community	https://tspace.web.att.com/communities/service/html/communitystart?communityUuid=3554733d-7fab-4a9f-a990-27d1231e94e9
 What and Why Microservices?	https://wiki.web.att.com/display/microservices/mS+Principles
 Platform Enabling TechDev	https://wiki.web.att.com/display/microservices/mS+Whitepaper
 “Get the Facts!” Webcasts!	https://wiki.web.att.com/pages/viewpage.action?pageId=555251459
 Wiki	https://operations.web.att.com/sites/ACSIRADAR/mSCatalog/Lists/mS%20Catalog/AllItems.aspx
 Microservices Catalog	https://operations.web.att.com/sites/ACSIRADAR/mSCatalog/Lists/mS%20Catalog/AllItems.aspx



Q & A



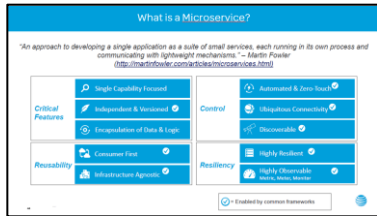
Appendix



Microservice Principles and Architecture



Microservices Principles



- ✓ Microservices are organized around **independent** business capabilities utilizing **domain-based design**
- ✓ **Decentralized data management** in which each service manages its own data
- ✓ High availability and scalability via **local** and **geo-diverse** redundancy
- ✓ Individual microservices are **easily replaceable** without impacting other applications or services
- ✓ Use **Direct Messaging Engine (DME)** for discovery based messaging
- ✓ Support **canary release** strategies to reduce the risk of introducing a new software version in production
- ✓ Delivery will be **Agile** & practice **DevOps**
- ✓ Quality assurance implements **100% automated unit testing** with > 90% code coverage
- ✓ Uses **Open Standards** where possible (TM Forum, GS1, etc)
- ✓ Purposeful design to **minimize AT&T-specific extensions**
- ✓ Microservices should be used directly **without proxies**
- ❑ AT&T will have access and update ability to all **source code***

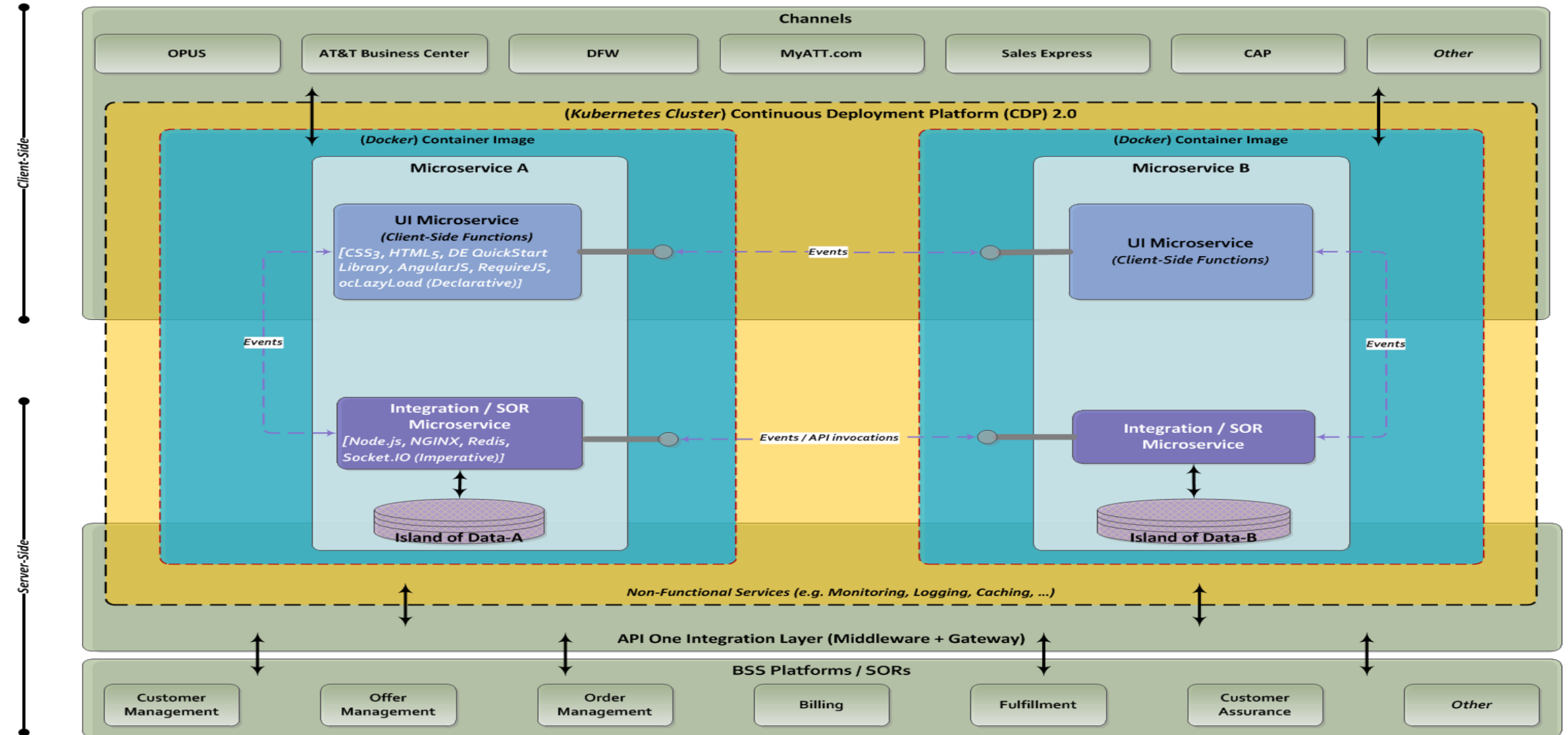
* Point of contention between Amdocs & AT&T



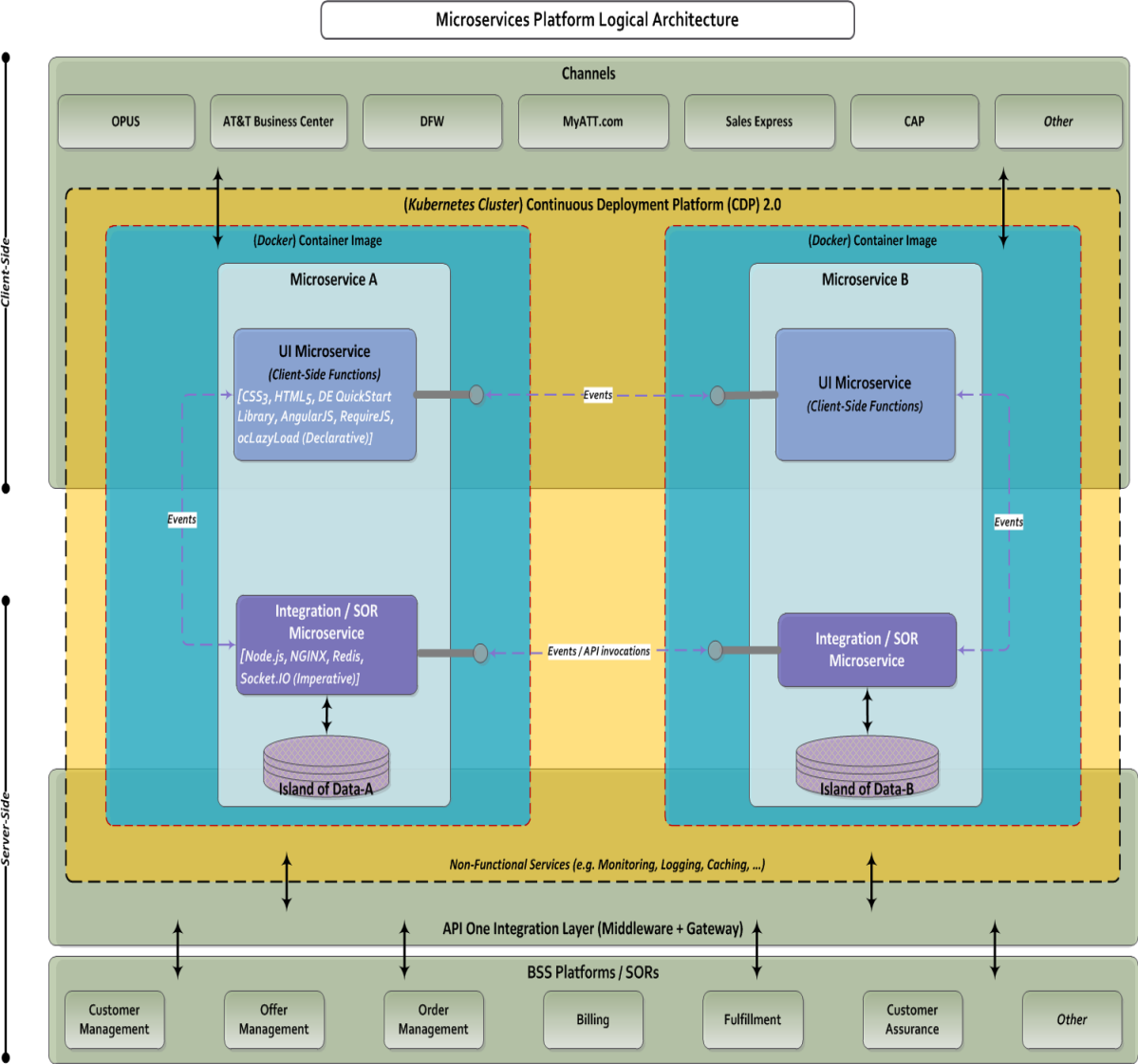
Microservices Platform – Logical View

The AT&T suite of Microservices deployed on a Common Platform

Microservices Platform Logical Architecture



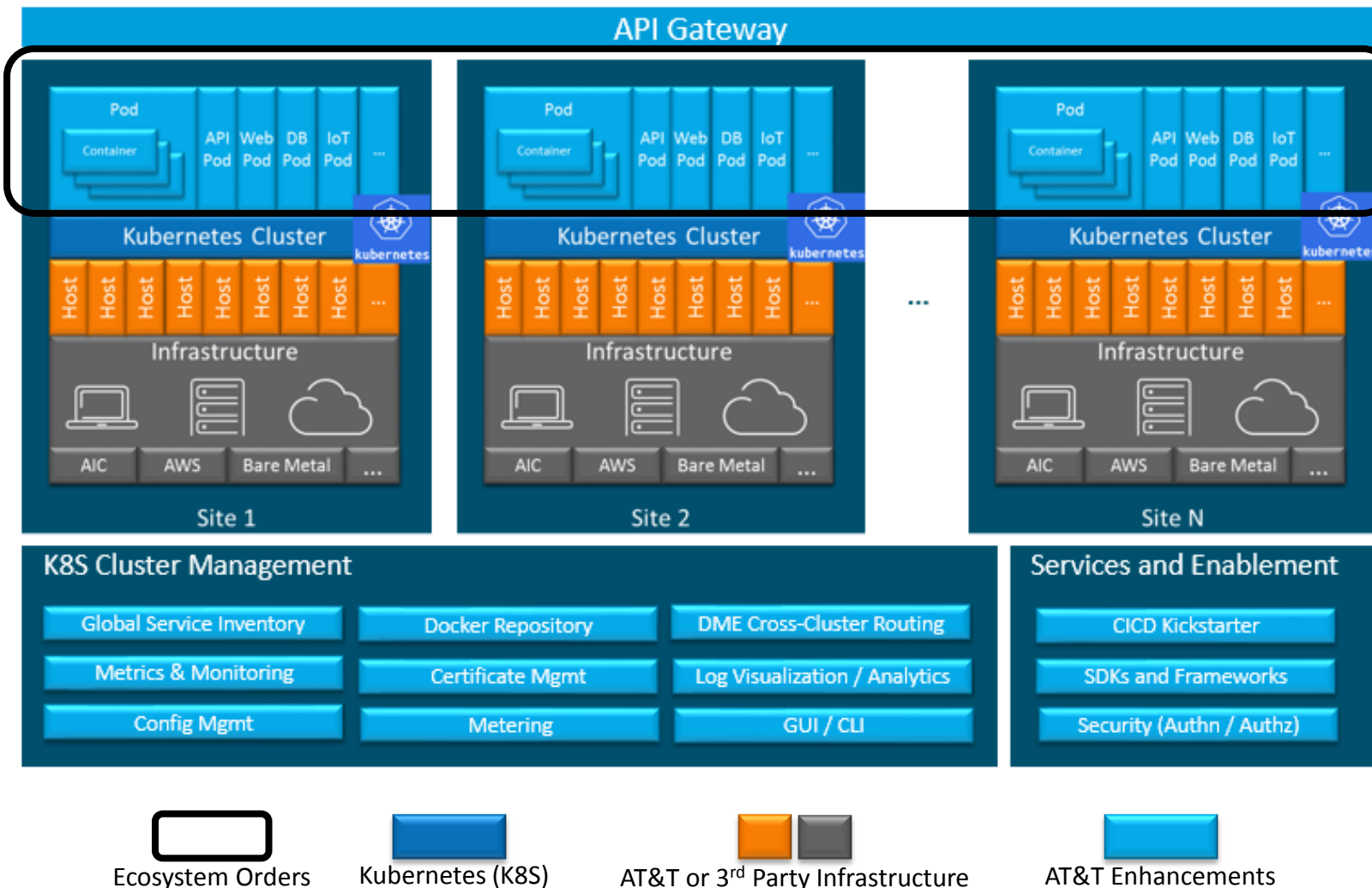
Technology Standards & Tools



Layers	Tools & Technologies
UI STYLEGUIDE <i>Defines User Experience Look & Feel</i>	1. CSS3
UI COMPONENTS & WIDGETS <i>Reusable UI elements - e.g. Buttons, Form elements etc.</i>	1. HTML5 2. DE QuickStart Library
UI FRAMEWORK <i>Reusable UI Code -e.g. Code for widget and widget container development</i>	1.1. ANGULAR JS 2. REQUIRE JS 3. OCLAZYLOAD
SERVER SIDE <i>Server Development - e.g. Services to expose SOR or persistency layer</i>	1. NODE 2. NGINX 3. REDIS 4. SOCKET.IO
PACKAGING TOOLS <i>Tools to package the microservice (web component and server side endpoints) to deploy near channel applications</i>	1. DOCKER
REPLICABLE CONTAINER PLATFORM <i>Container provides non functional services such as monitoring, logging and caching for the microservices.</i>	1. DOCKER 2. KUBERNETES



Microservices Platform – Physical View



STATUS SUMMARY

Completed

- Initial Hardware Orders submitted; pending funding
- Framework identified
 - Hardware: AIC or EC Cloud
 - Technology: AT&T OnePlatform
 - CI/CD : Kickstarter / Pipeline
 - Software: Java IDE (Eclipse)
- Established an initial development environment
- Established basic, manual deploy cycle

Activities in Progress

- OnePlatform support model
- AIC Host or Tenant Managed Model
- Capacity delivery for 1Q microservices
- AJSC6 - mitigated via current OnePlatform
- Data Platform selection per microservice



Domain Analysis, Scoring & Prioritization of Microservices

4-Step Process – from Domain Analysis/Decomposition to Roadmap/Execution

STEP 1: Domain Analysis & mS Candidate Identification*

- Individual teams (ABS, EG, DE) perform the domain analysis and identify the mS candidates ('right sizing'); analysis may be done 'top-down' or 'bottom-up' to derive the right-sized candidates
- E.g. Offer/Product domain owner will identify the mS candidates in that domain via the domain analysis process

* ABS, EG, DE will collaborate closely to eliminate duplication of mS candidates, and identify opportunities for reuse of the mS candidates across the business units

STEP 2: mS Patterns Assessment

- A small team (within each of ABS, EG, DE) will assess the mS candidates to determine if they adhere to the prescribed mS standards, best practices & design patterns
- Appropriate SMEs will assist to clarify on the purpose, structure and other aspects related to the mS candidate as necessary

STEP 3: mS Prioritization

- The domain owner (within each of ABS, EG, DE) performs the mS scoring process using the common scorecard model
- Projects that have PIDs may not be scored since they are already funded
- New mS candidates will be scored by the domain owner, and will be prioritized based on the scores

STEP 4: Roadmap Development

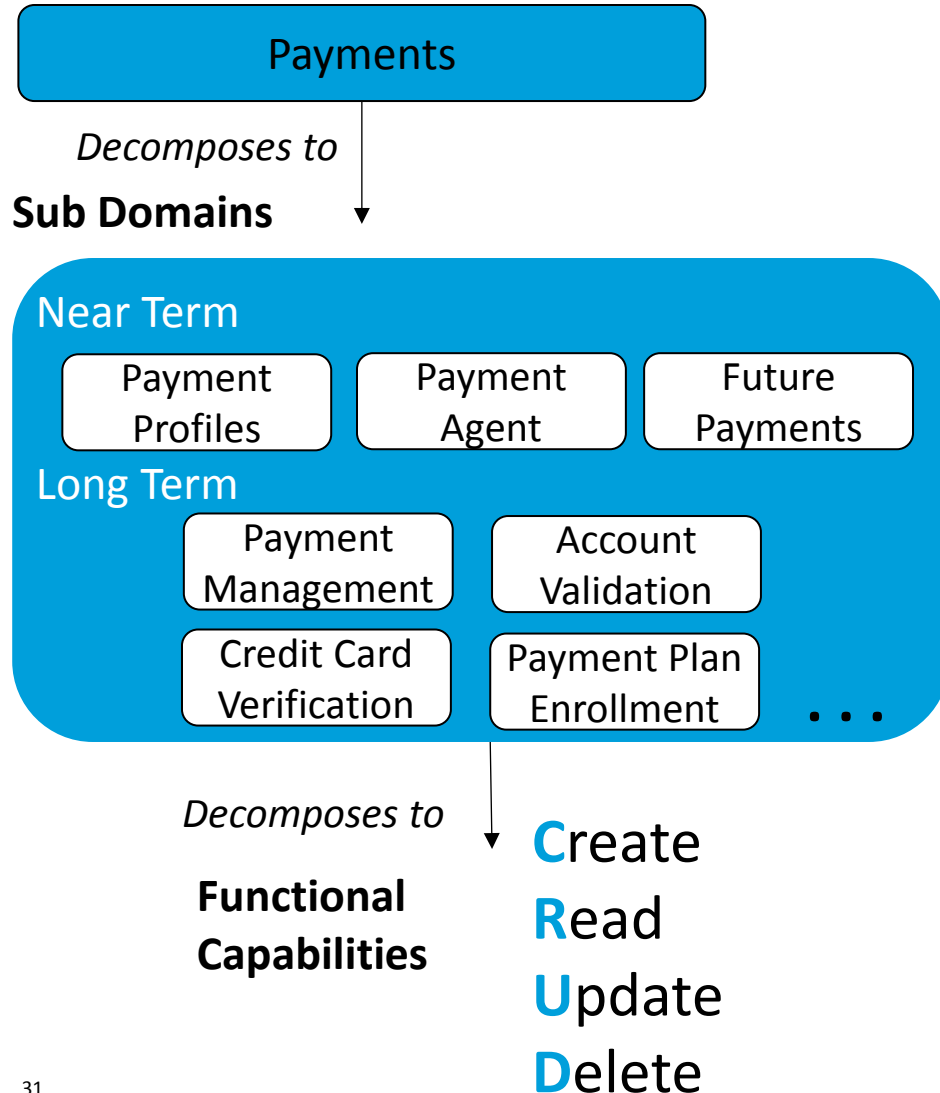
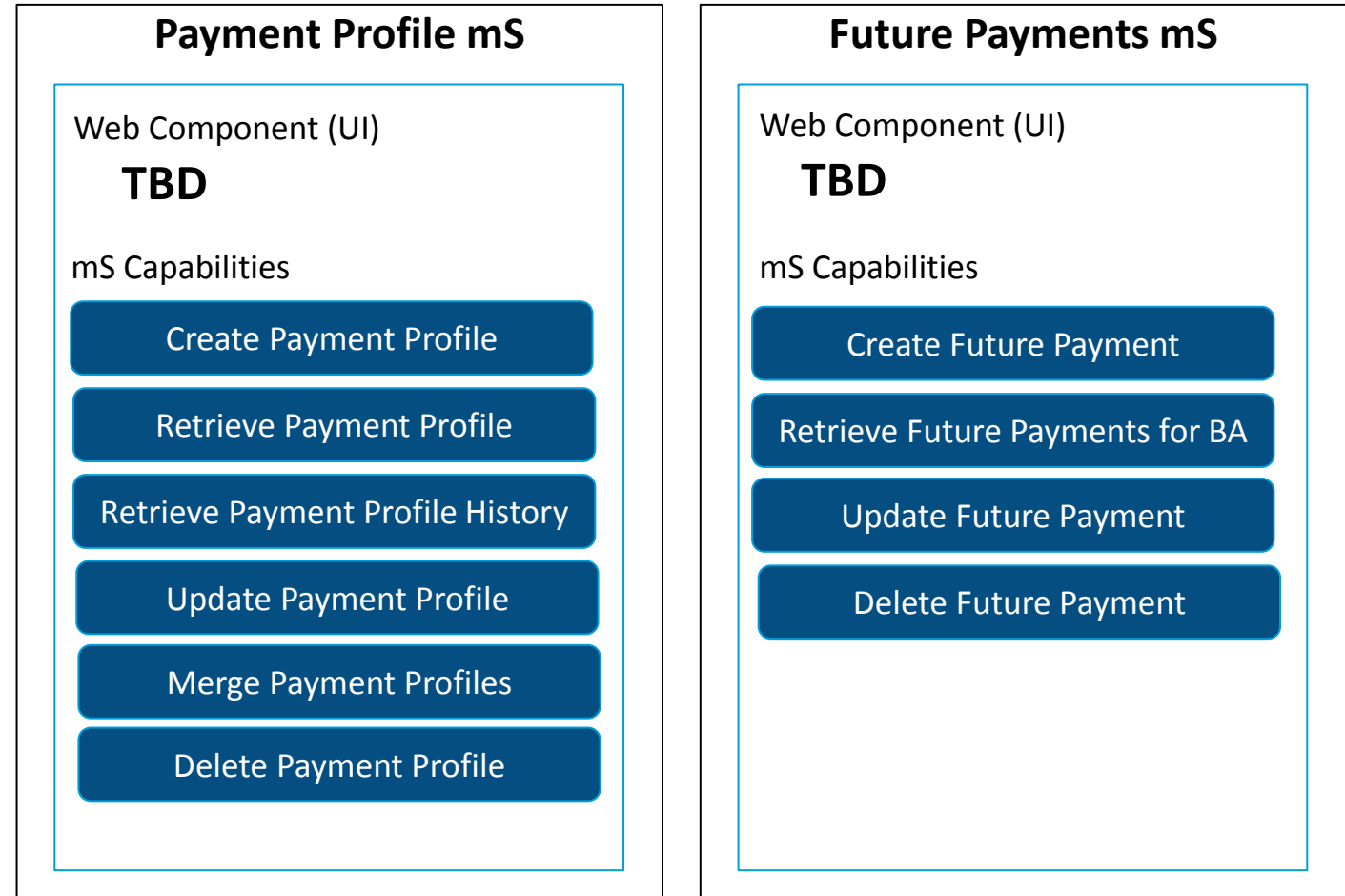
- mS roadmap leads (from each of ABS, EG, DE) will inspect the scores for all of the mS candidates across ALL the domains
- Overall mS scores will be used as input for prioritization, construction of the ABS/EG/DE mS roadmaps, and assignment to projects for execution

Microservice Use Cases & Metamodel



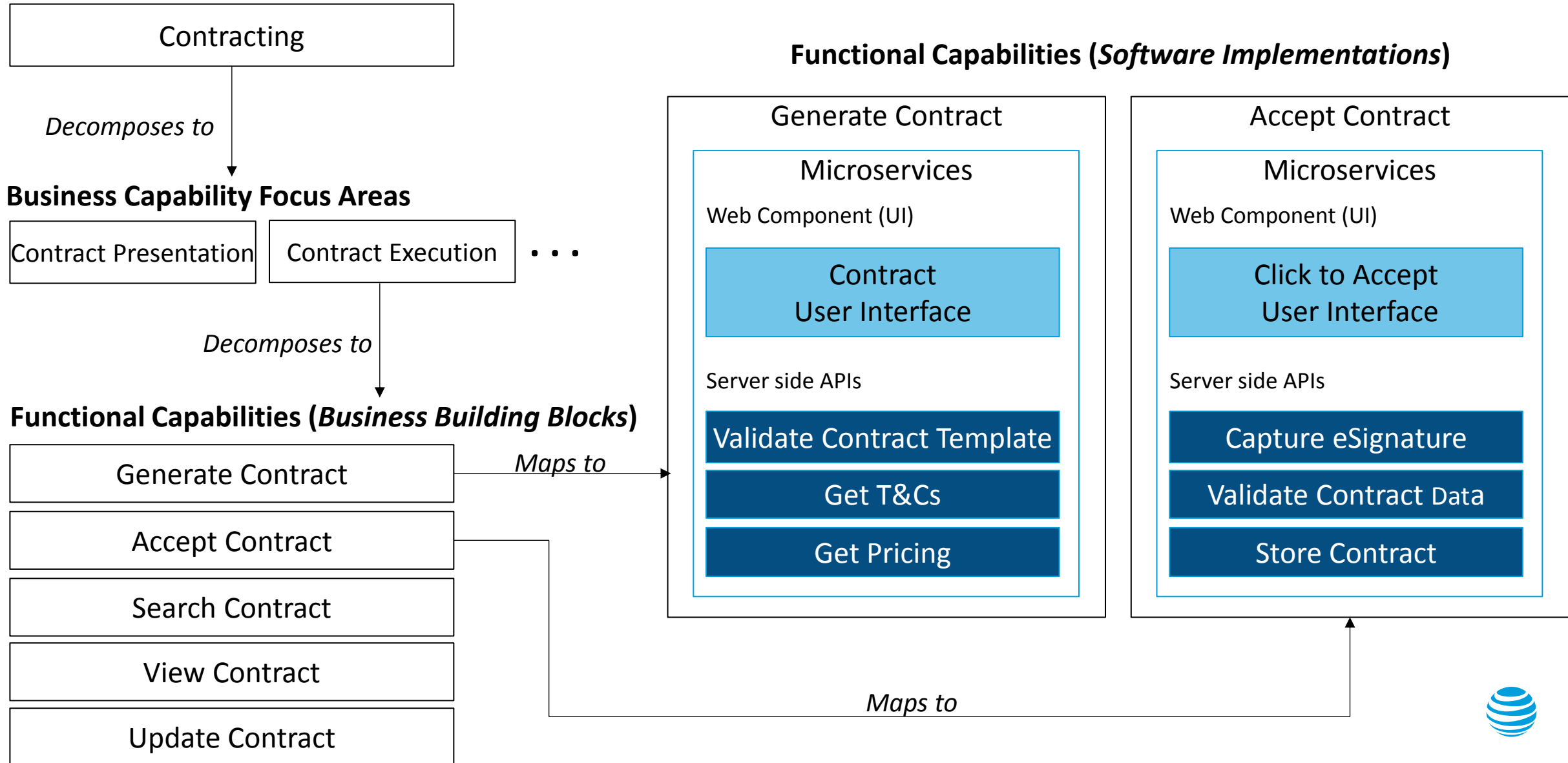
Sample Use Case – Payments Suite

Business Capability

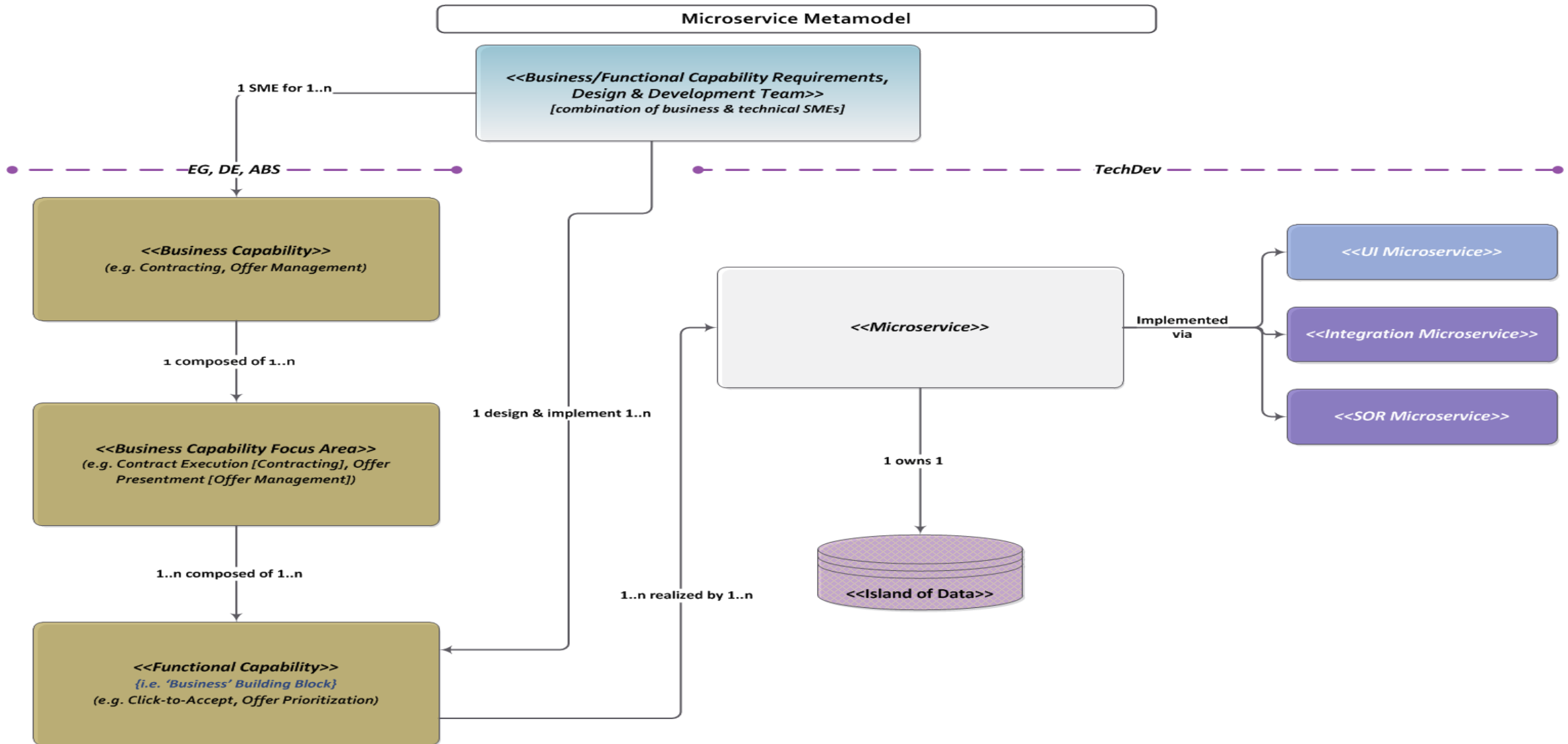
Functional Capabilities (*Software Implementations Examples*)

Microservices – Contract Management Use Case

Business Capability



Microservice Metamodel



Metamodel Element Definitions (1)

Business Capability (*ref. Alfabet*): A business capability defines the organization's capacity to successfully perform a unique business activity. Capabilities:

are the building blocks of the business

represent stable business functions

are unique and independent from each other

are abstracted from the organizational model

capture the business' interests

Business Capability Focus Area: Finer-grained '*areas of focus*' that higher-order business capabilities are composed of.

Functional Capability: Finer-grained '*business building blocks*' that higher-order business capability focus areas are composed of (or assembled using). Functional Capabilities are also reusable units of code that are product-agnostic, and that may be used by multiple business solutions.

Microservice: An architectural style in which applications are composed from loosely coupled API services with automated & independent lifecycles.



Metamodel Element Definitions (2)

UI Microservice: A type of microservice used to implement UI/UX capabilities for a microservice-based application. The UI microservice uses reusable widgets to implement UI/UX for a variety of client devices such as phones, tablets, kiosks, and STBs in addition to a standard web interface.

Integration Microservice: A type of microservice that uses other backend microservices, APIs or web services to access business logic, data fabric interfaces, enterprise workflow engines, business rules engines, etc. and integrate the same to deliver the result to the calling client. The calling client in turn may be an edge application or a UI microservice.

SOR Microservice: A type of microservice that is specifically implemented to access a SOR for data that may be requested by other microservices, APIs or web services. The SOR microservice limits the access to the data that it manages, and hence provides data encapsulation.

Island of Data: Data and data structures specific to a given microservice; could be a logical cache fragment, but is NOT a SOR.



Microservice FMO (Proposed)



Principal Microservices Domains (EG)

OMS Decomposition will serve as a starting point towards discovering microservices for use across several business domains in AT&T

Order Management (OMS Decomposition)

Offer & Product Management

Consumer Sales

Payment Processing

Analytics & Reporting

Customer Service

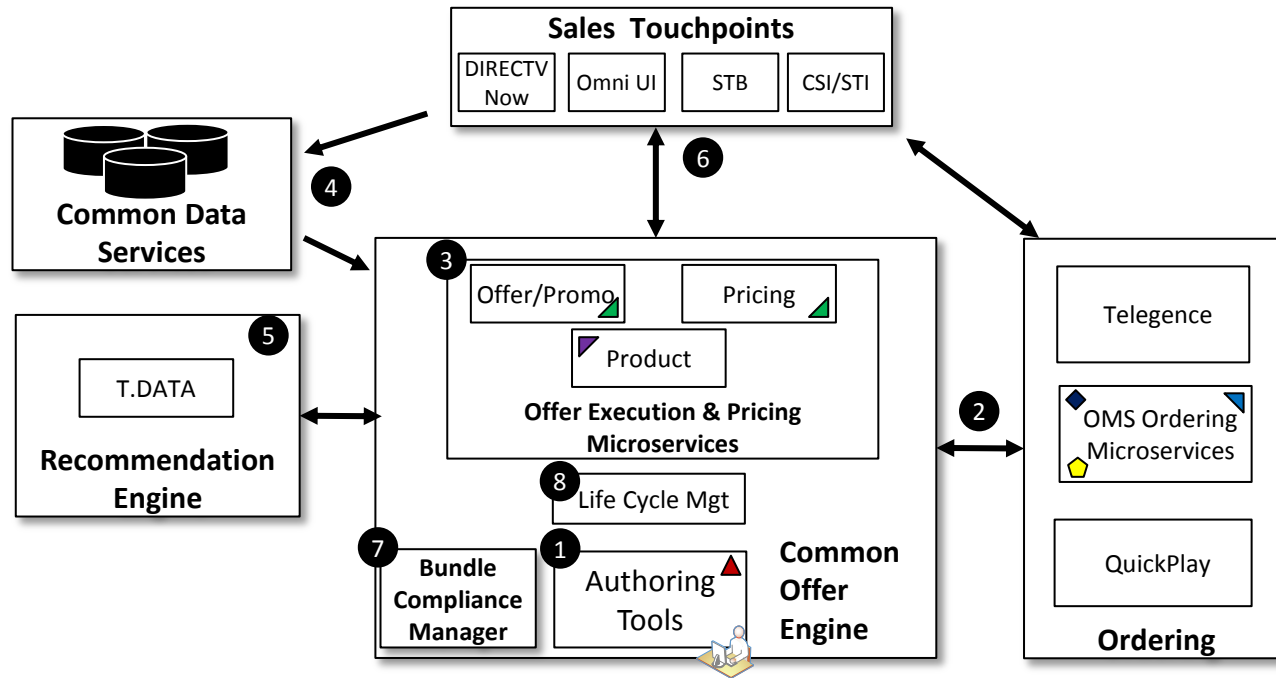
Technical Care

Billing

...



Common Offer Engine – Long-Term Vision



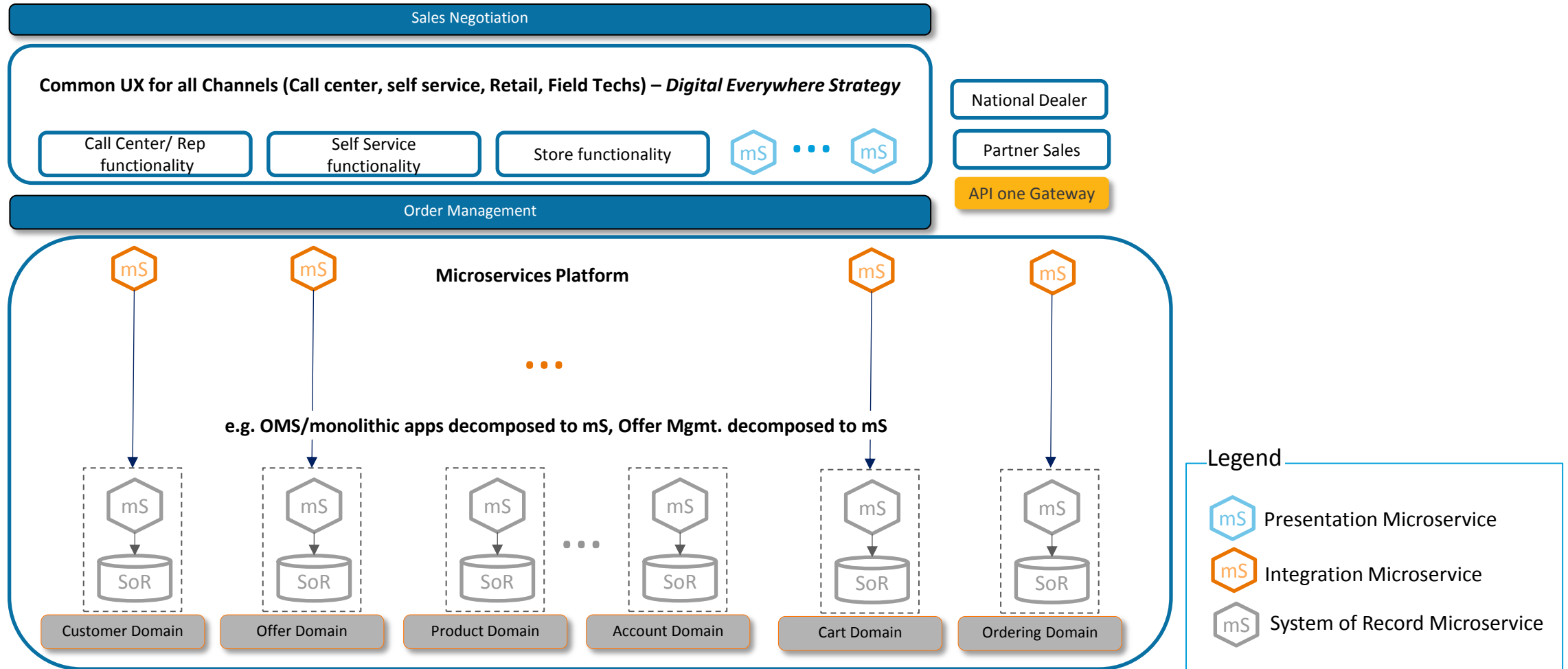
- **Centralized Product & Offer Management**
- **Common High Performing Microservices**
- **Decoupling of Offers from Ordering & Billing**

1. Authoring tools to allow for centralization of product, pricing, offer & promotion configuration. Enables the business to respond to changing markets.
2. Synchronization of data needed for ordering and billing across multiple systems (transitional)
3. Common set of microservices with high performance caching and rules execution to meet the demands for online, STB & more, decoupled from ordering & billing
4. Integration with common data services to determine eligibility, link accounts and track offer lifecycle
5. Integration recommendation engine for offer prioritization
6. Execution of point of sale eligibility, listing of offers, pricing & quotation, upfront notification of potential offer breakage
7. Systematic evaluation of offer compliance based on account triggers, right sizing of accounts when compliance is broken and notification of the account change.
8. Life cycle management of products, pricing, offers and promotions

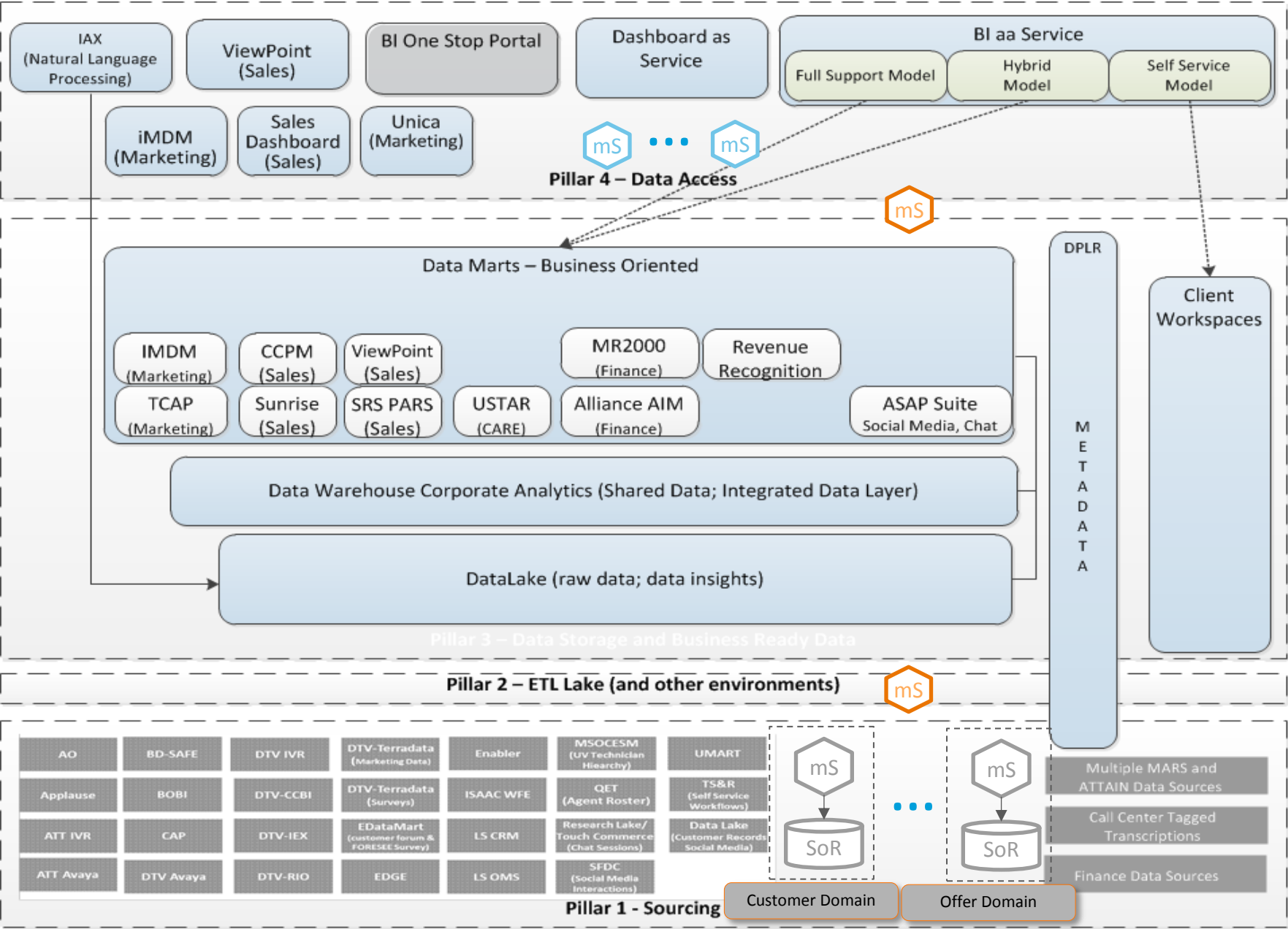


Consumer Sales FMO – Long Term Vision

Omni Channel Vision – Enabled by Digital Everywhere strategy and pivot to microservices from monolithic applications



Analytics & Reporting – Long Term Vision



MOBILIZING
YOUR
WORLD™

