

Digital Transformation: New Image through a Modern API Platform



Raymond Perry

Senior Manager, AVP

Enterprise Integration & Strategy



An accomplished IT Technology Leader and Enterprise Architect that specializes in enterprise integration, technology strategies, new technology, engaging external vendors, and insuring development teams deliver solutions that meet or exceed business expectations. Has a proven track record of providing thought leadership, strategies and roadmaps in Integrations, SOA, and API domains for the enterprise and business partners.

– 28 Years in the Technology Industry

- Finance
- Wealth Management
- Retail



UMB Financial Corporation (Nasdaq: UMBF) is a financial services company headquartered in Kansas City, Mo. UMB offers **PERSONAL BANKING, COMMERCIAL BANKING, HEALTHCARE SERVICES and INSTITUTIONAL BANKING**, which includes services to mutual funds and alternative-investment entities and investment advisory firms. UMB operates banking and wealth management centers throughout **Missouri, Illinois, Colorado, Kansas, Oklahoma, Nebraska, Arizona and Texas**. For more information, visit UMB.com, [UMB Blog](#) or follow us on Twitter at @UMBBank, [UMB Facebook](#) and [UMB LinkedIn](#).

Founded in 1913

Zilker Technology at a Glance



Consulting,
Implementation
Services, Managed
Services

Niche Skills of a
Boutique
Consulting Firm

Experience of Global
Integrator

What is Digital Transformation?



Digital transformation represents a break from the past, with a high level of impact and complexity, Transformation efforts need to be undertaken as the means of getting to a differentiated capability to accomplish a defined goal.

Pearl Zhu



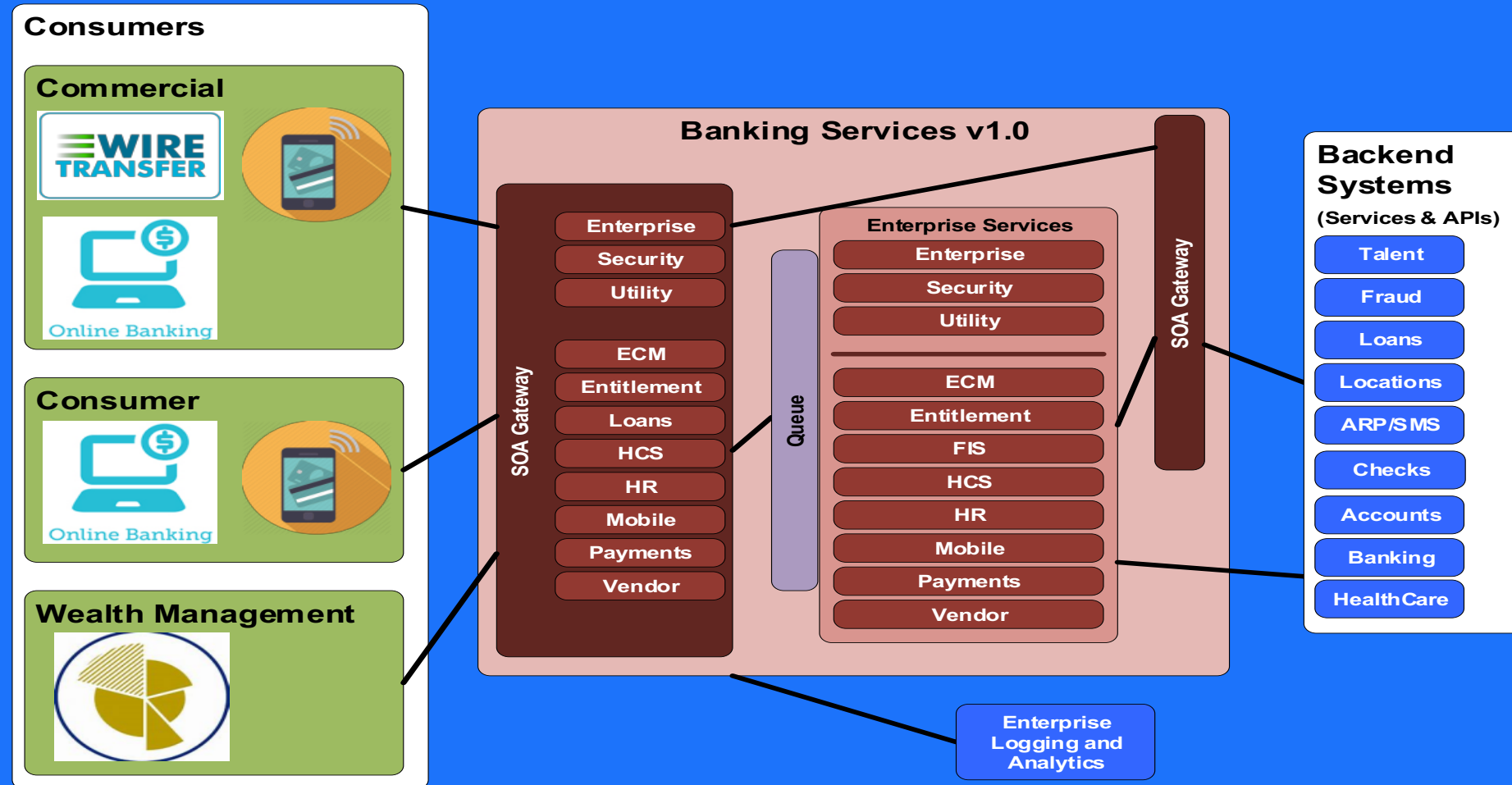
Banking SOA Gateway



1. Enterprise Service Bus
2. One Connection Point
3. Security Considerations
4. Banking Services

But....

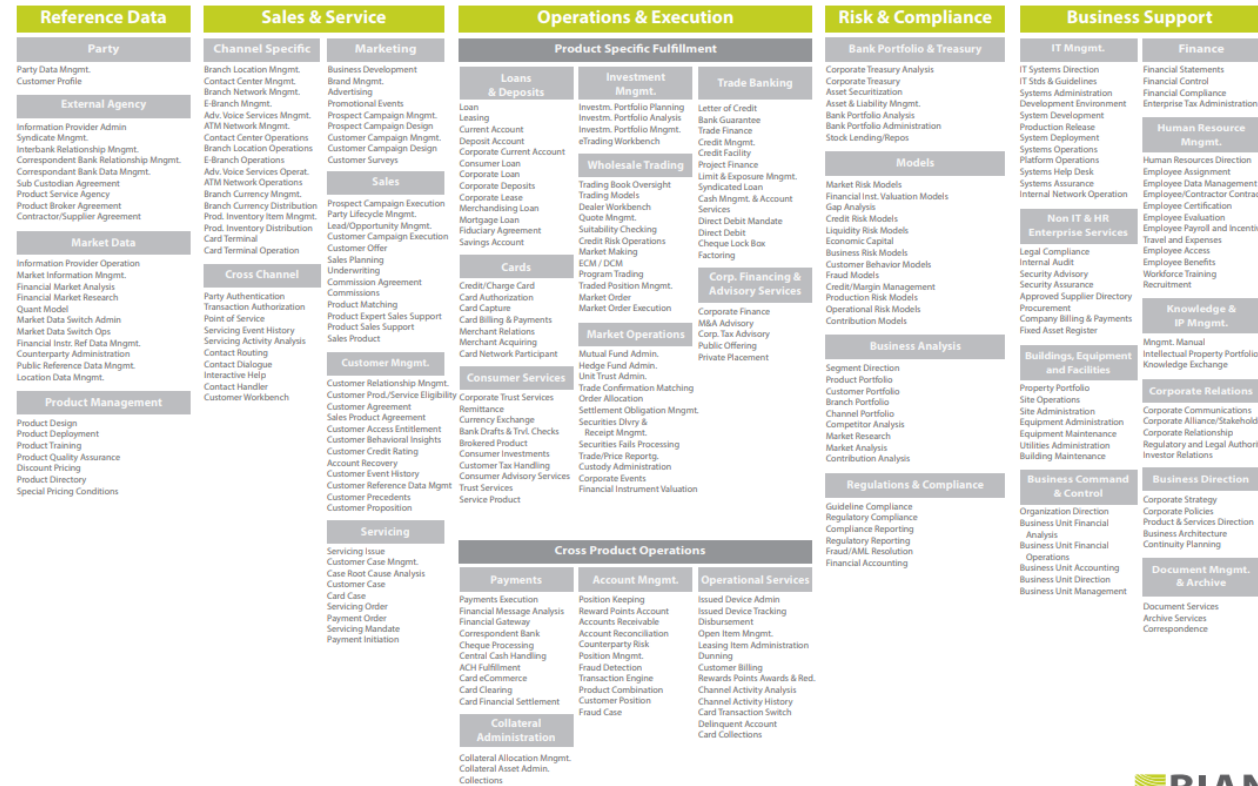
1. Application Services
2. Point to Point
3. Limited Re-use
4. No Catalog
5. Limited Client Knowledge



Industry Concepts – BIAN as an example



BIAN Service Landscape - V7.0



Copyright © BIAN 2018



Helps us to know what to provide as an API Owner

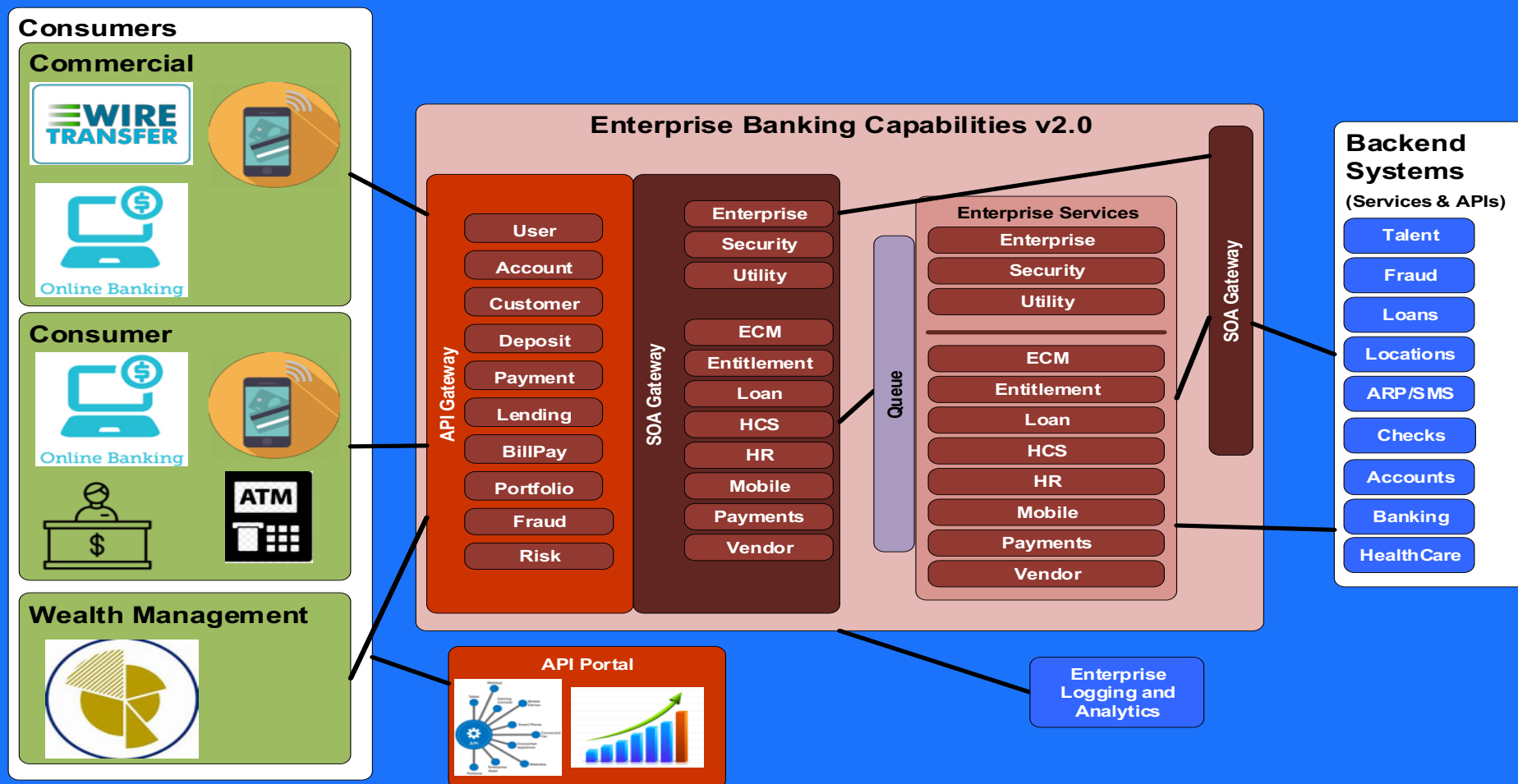
Helps our partners to know what to expect as an API Consumer of our services

Helps us to know what to expect as an API consumer of other companies' services

Banking API Gateway



1. Banking Capabilities
2. Canonical Model
3. Re-use of Services
4. Catalog of APIs
5. Knowledge of Clients
6. Additional Security
7. Promote APIs



Capability Framework

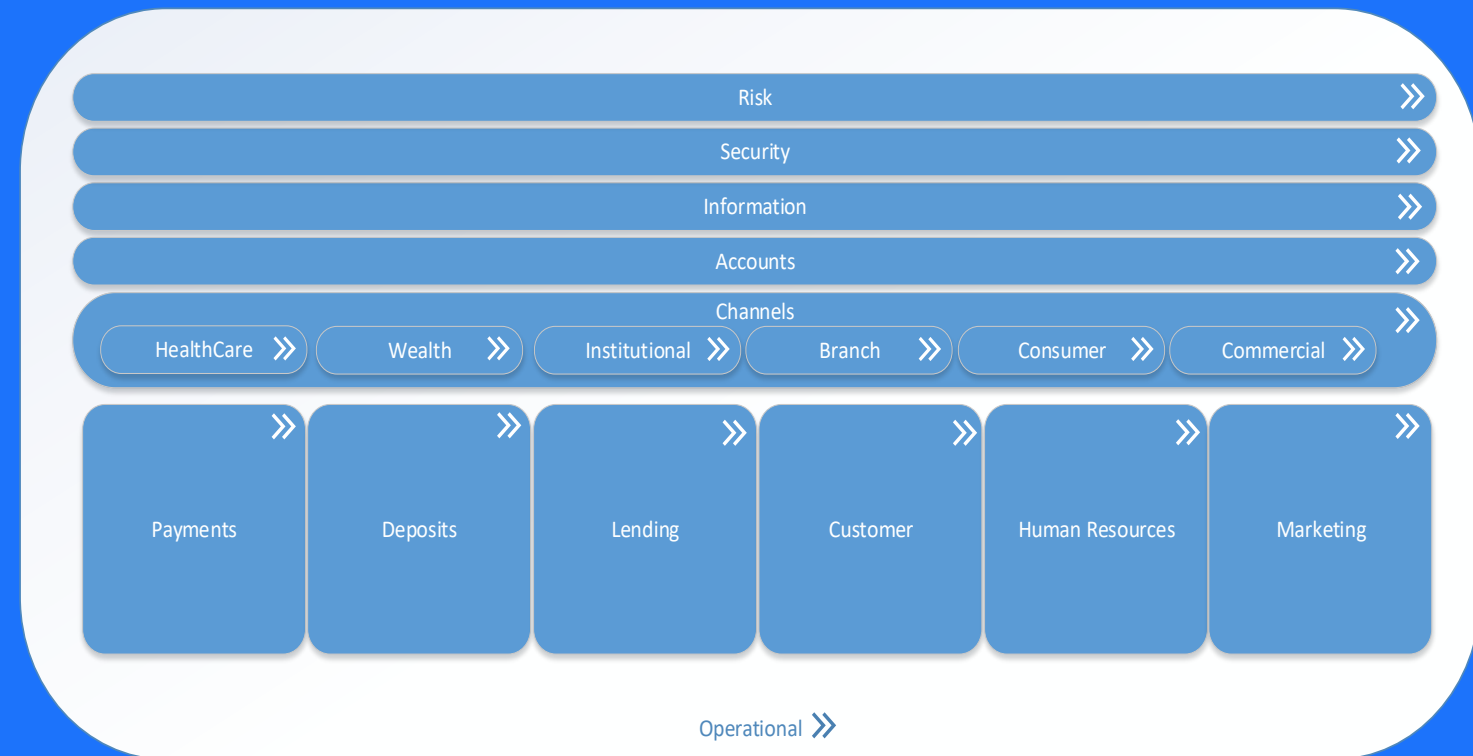


■ The Capability Framework is the Glue!

- ✓ To illustrate a starting and ending point for success
- ✓ To understand how best to reuse domain intelligence
- ✓ To break large concepts to small ones

Align Capability Framework with BIAN Model

BIAN = Banking Industry Architecture Network



Capability Framework - Domain



Last Update: 4/4/2019

- Domains break down to capabilities

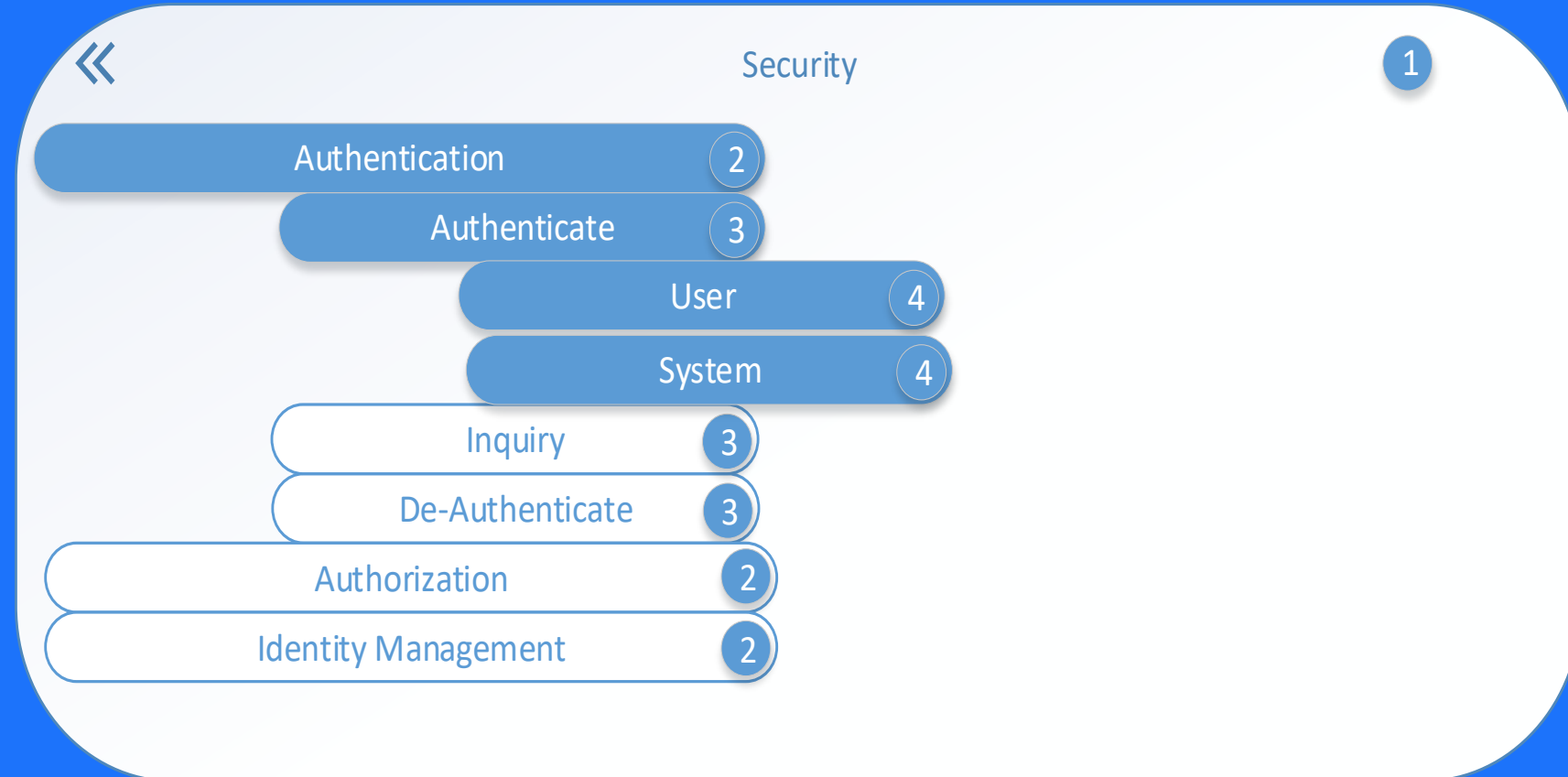


Capability Framework - Capability



Last Update: 4/4/2019

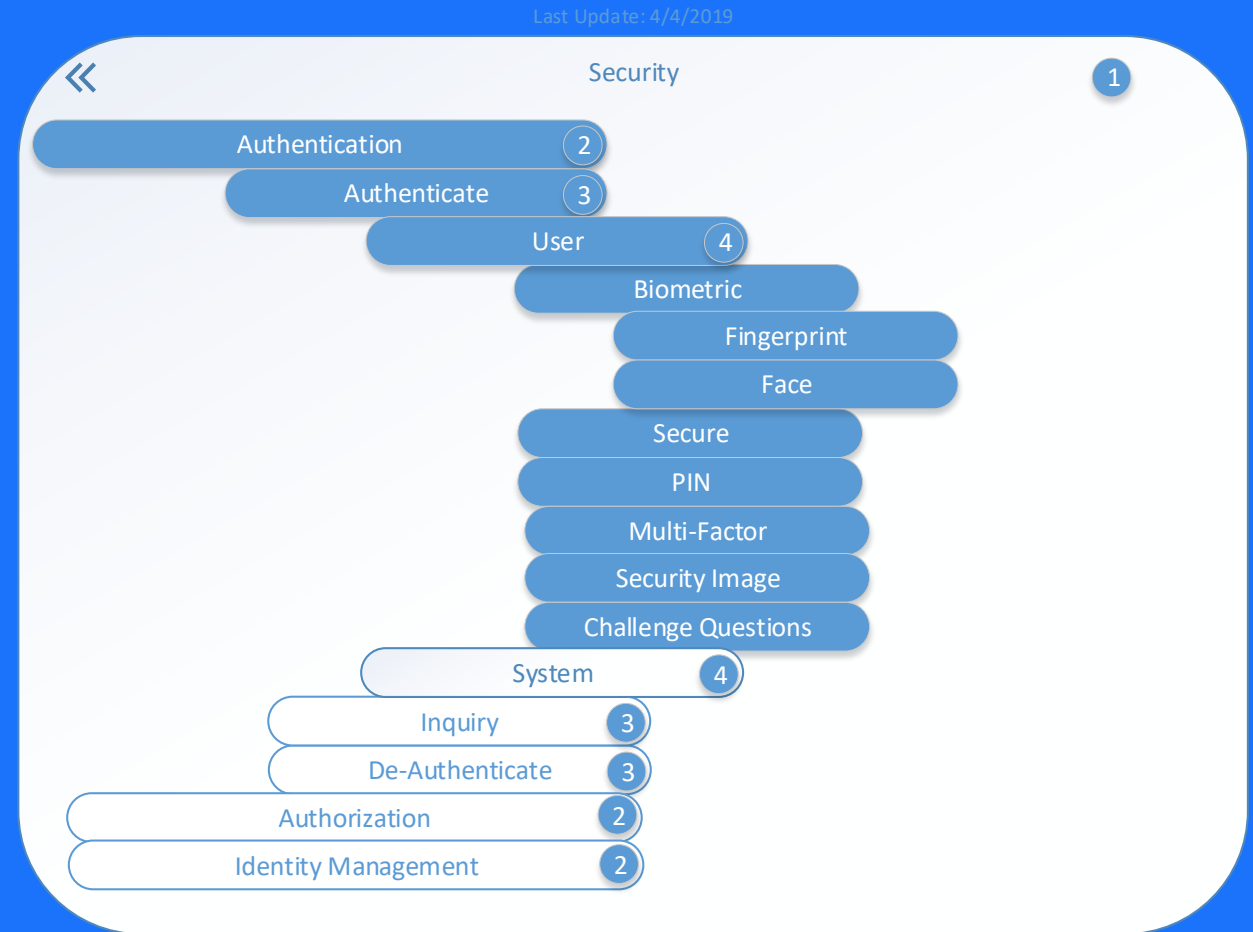
- Capabilities break down to a category



Capability Framework - Capability



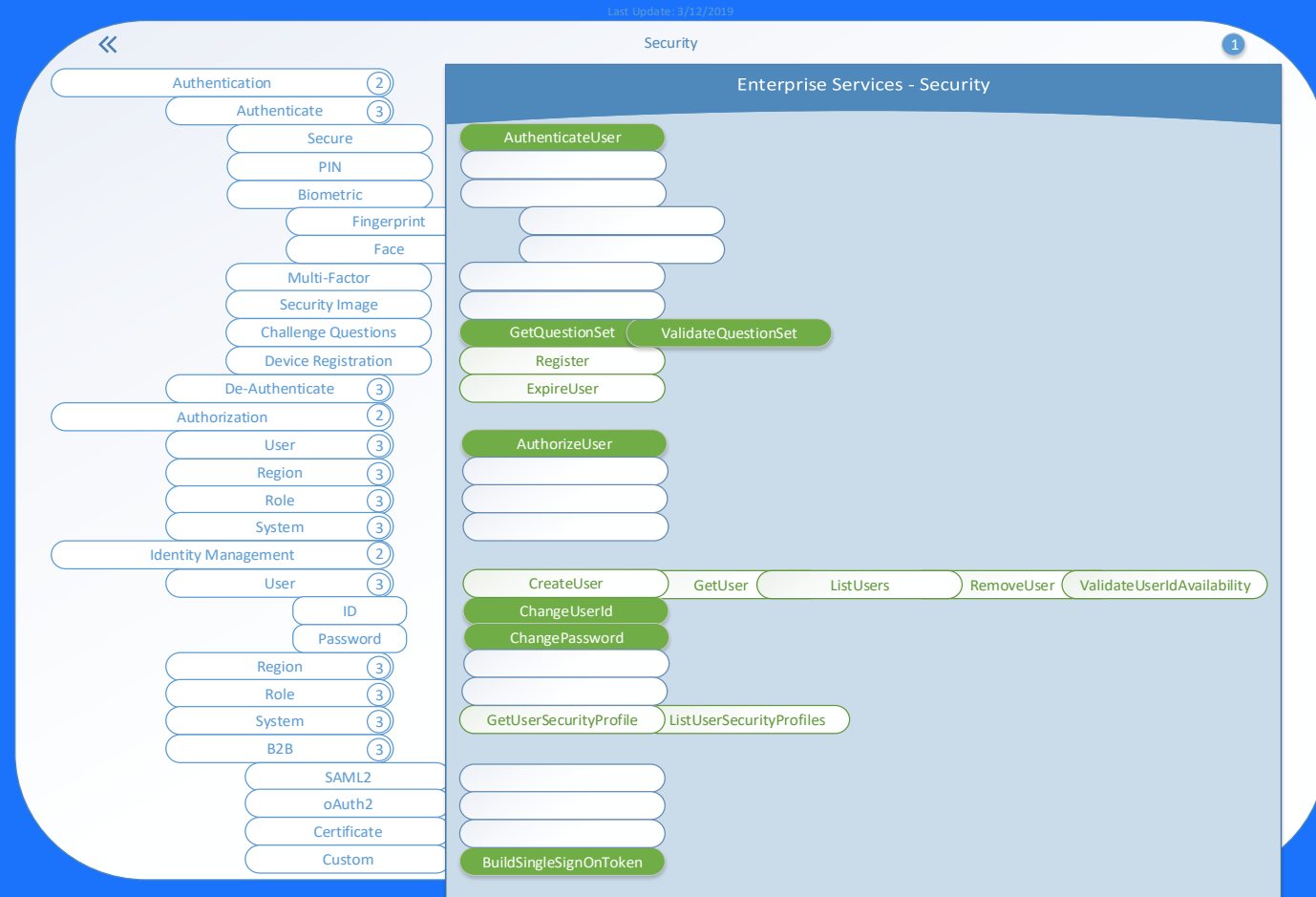
- We can continue to break down and refine work



Capability Framework - Capability



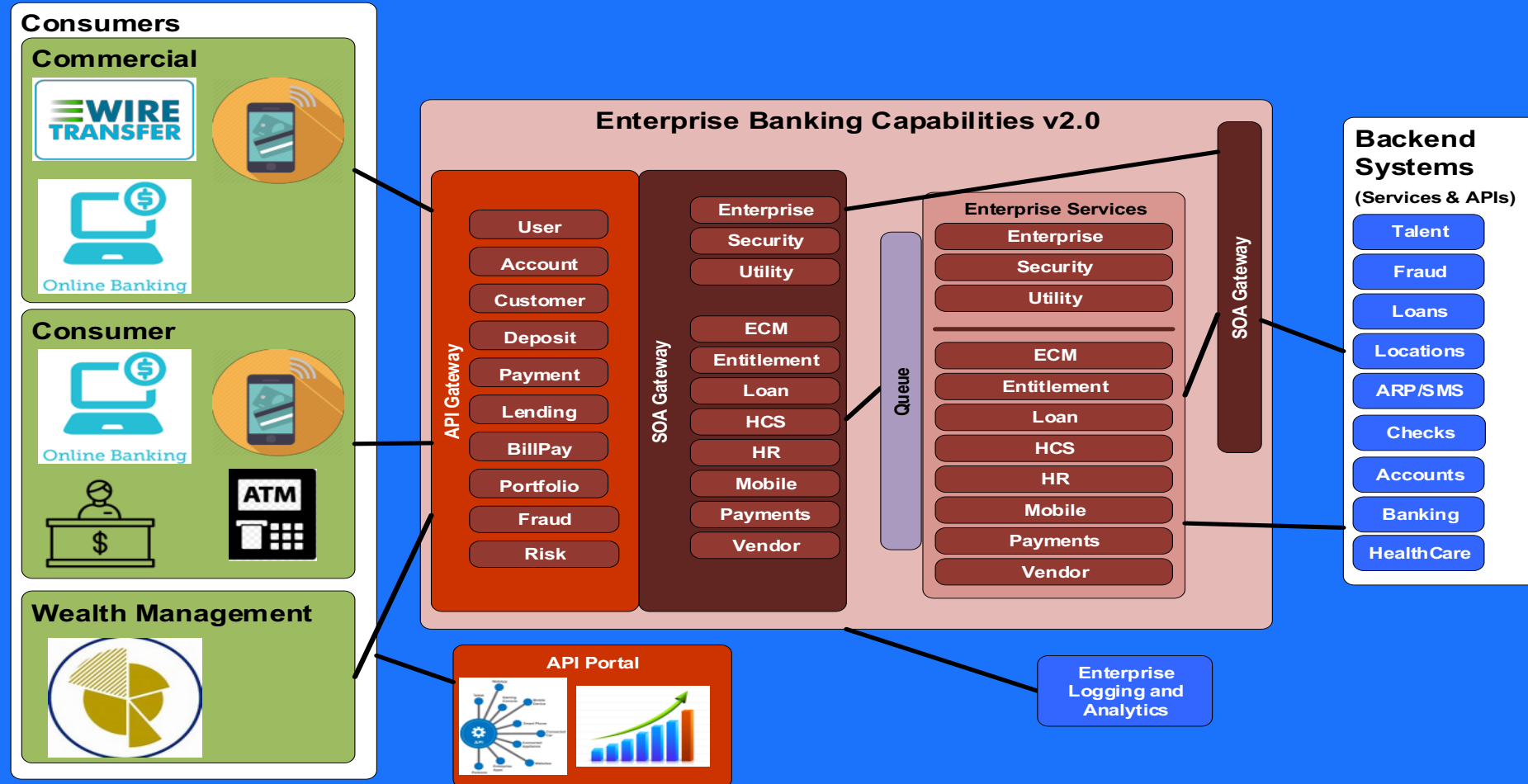
- Break it down to the smallest component.
- End result matches swagger and models.



Banking API Gateway



1. Banking Capabilities
2. Canonical Model
3. Re-use of Services
4. Catalog of APIs
5. Knowledge of Clients
6. Additional Security
7. Promote APIs



5 Key Take a Ways



1. Plan your Design
2. Plan your Delivery
3. APIs First
4. Version
5. Plan for Production



Plan Your Design



1. Define your canonical model
 - Based on Capabilities
 - Consider Source Systems
 - Don't consider presentation
 - Build Micro-ish Services
2. Define your APIs
 - Based on you canonical model
 - Consider your consumer applications
 - Presentation shouldn't be considered
3. Use the Tool you plan to deliver with
 - Build your swagger from your delivery tool



Plan Your Delivery



1. PLAN

- Design API
- Services vs API's
- Realistic estimates
- Match demand with Capacity
- Speed compromises quality

2. ORGANIZE

- Follow model and capabilities
- Deliver By: Capability or Product Line
- Delivery Complete Service/API
- Plan major enrichments with major version

3. TEST

- Define and build re-usable test scripts
- Consider Environments
- Reduce defects



1. Deliver APIs
 - Deliver APIs before Usage
 - Organized Delivery
 - Meet your sprint dates
 - Reduce Noise
2. Expect Business APIs
 - Business Specific Formatting
 - Deliver what is promised
 - Presentation can be considered
3. Plan Testing
 - Application testing
 - Stubbed services



API Versions



1. Light-weight Governance
 - Governance Overkill
 - Keep it simple and workable
2. Plan for the lifecycle
3. Plan to handle mix versions
4. Communicate release notes
5. Plan Consumers upgrades
6. Version Support



Plan for Production



1. Access and Permission

- Access to Portal
- Access to Gateway
- Access to APIs
- Communication Security

2. Protect Your Data

- Internal
- External
- Different API response payload per consumer or consumer type

3. Support

- Portal
- Gateway Execution
- On-call

4. SLA's & Analytics & Dashboards

- Reports that will be distributed
- Specific reports for banking units (limit the noise)



Thank You

