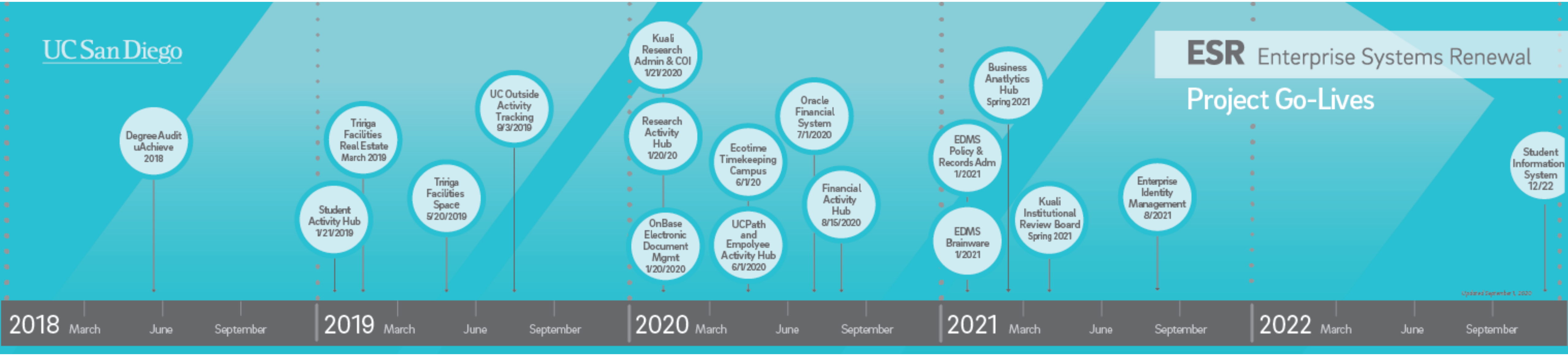
UCSD Integration and Data Strategy

Brian DeMeulle

Scott Lee

Ashish Pandit

Near term: ESR Program – Timeline & Scope



Renewal Agenda

- Mainframe retirement: Student (RFP in progress), Finance (Oracle), Research (Kuali) and Employee (PeopleSoft)
- Modernize “Fit for Purpose” applications: Degree Audit, Facilities, Time Keeping, Learning Management, EDMS, etc.
- Activity Hub for blended data
- Middleware for data integration

Big Rules

- Cloud first
- Open Source wherever possible
- Free the data (data democratization – accessible to the average user)

Long Term: Digital Transformation Playbook

Better

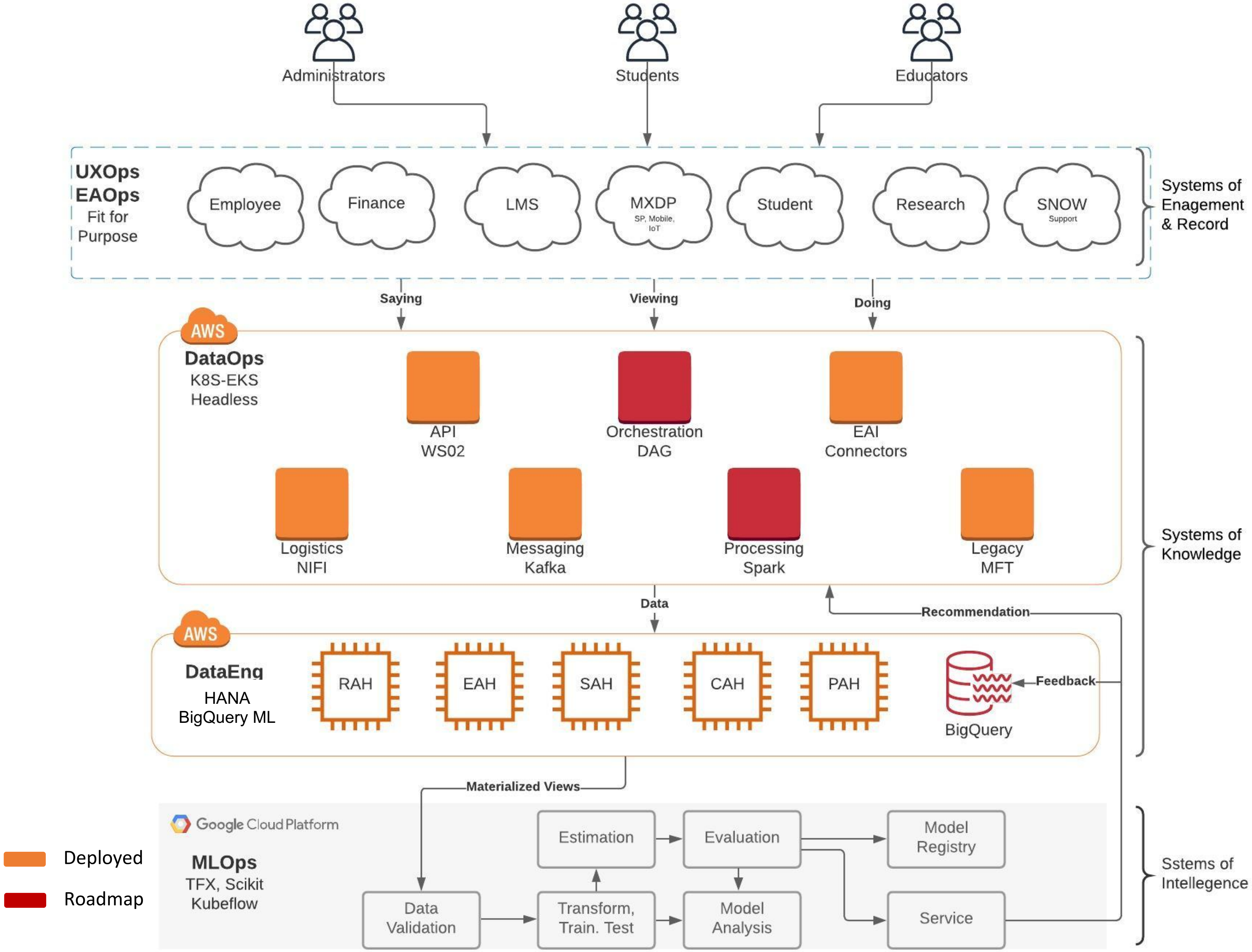
- Speed of life
- Contextual Knowledge
- Next Best Action

Faster

- Time to value
- Seize opportunities in the marketplace
- Deal with disruptions to mission

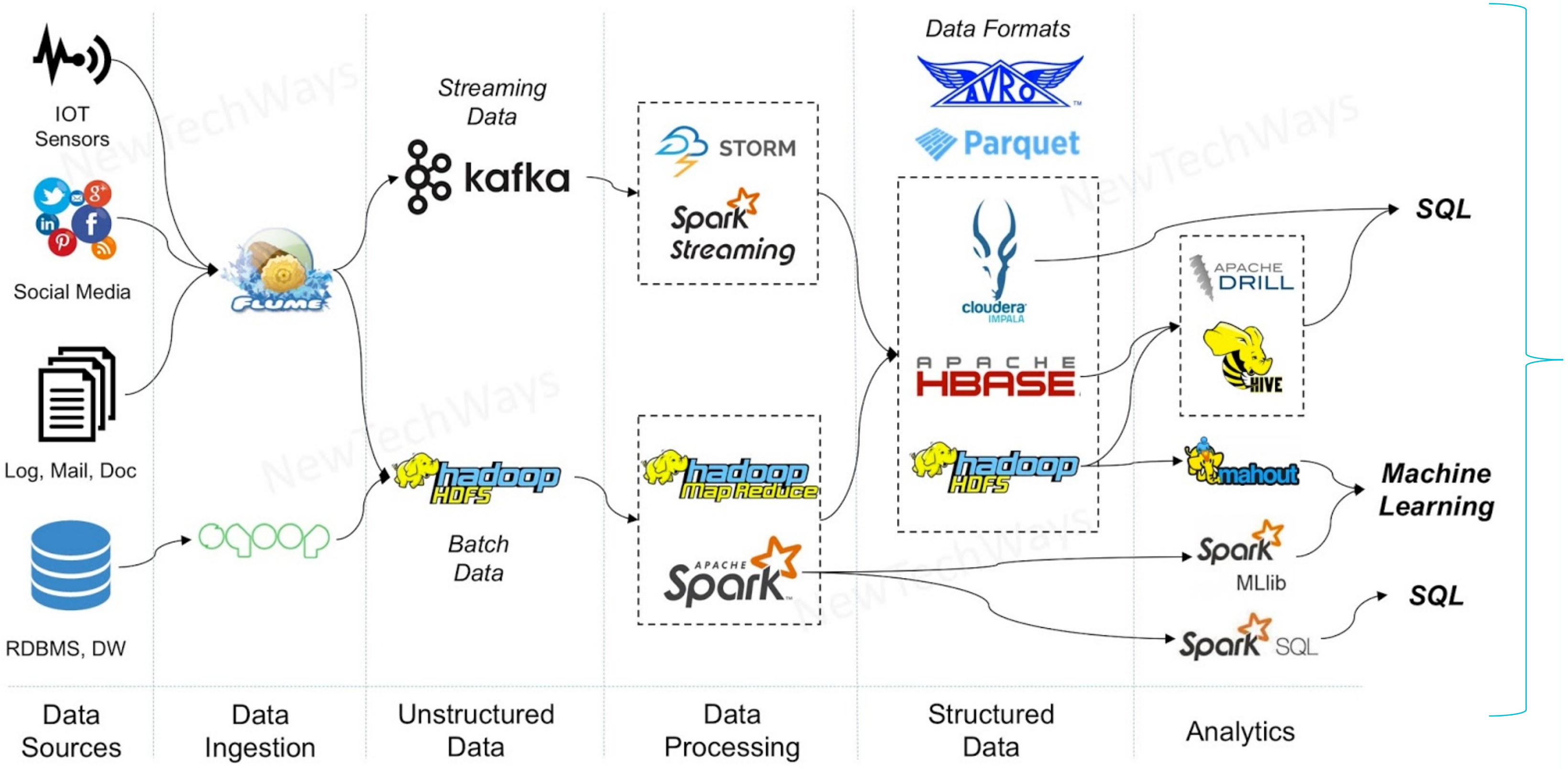
Cheaper

- Just in time deployment
- Less manimation
- True detect and respond operating model



What is streaming?

Event stream processing (ESP) platforms are software systems that perform **real-time** or **near-realtime** calculations on event data "in motion." - Gartner



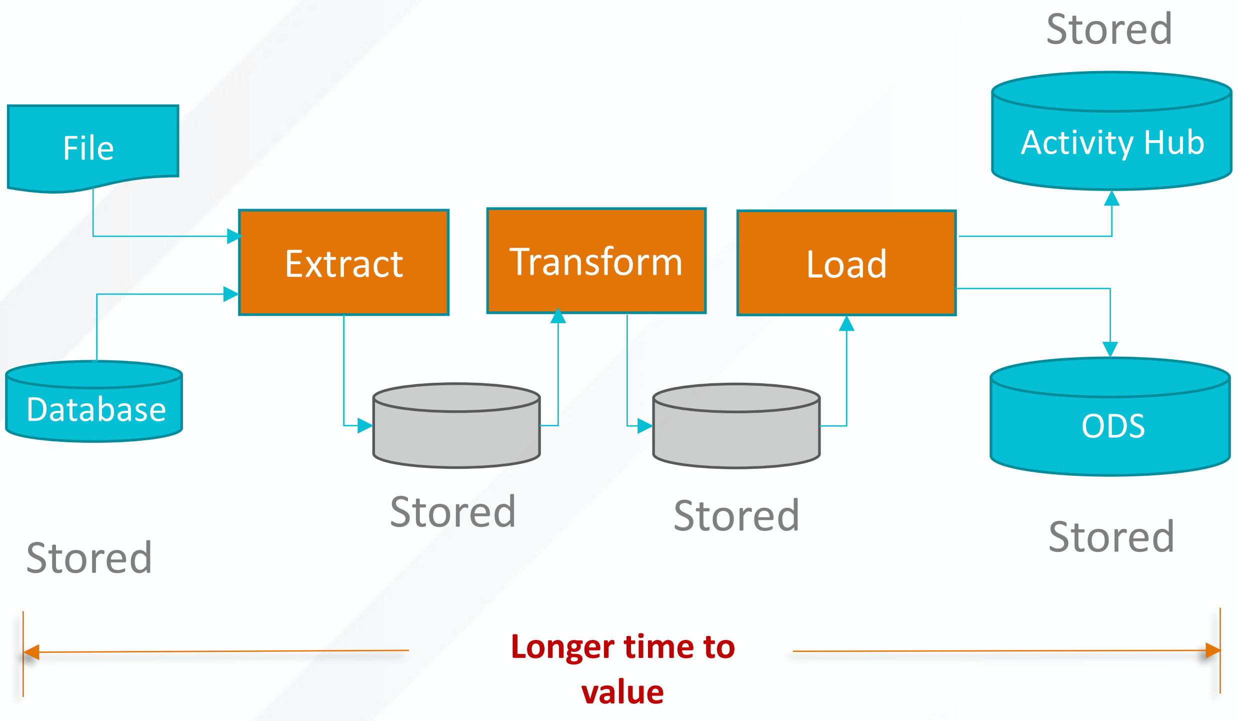
Why? Increasingly demanding consumers are pushing analytics from **transactional** to **continuous.**

Moving from information to **(contextual) knowledge or insights**!

It’s why many of the tools have been developed for the data science community e.g. **data pipelines.**

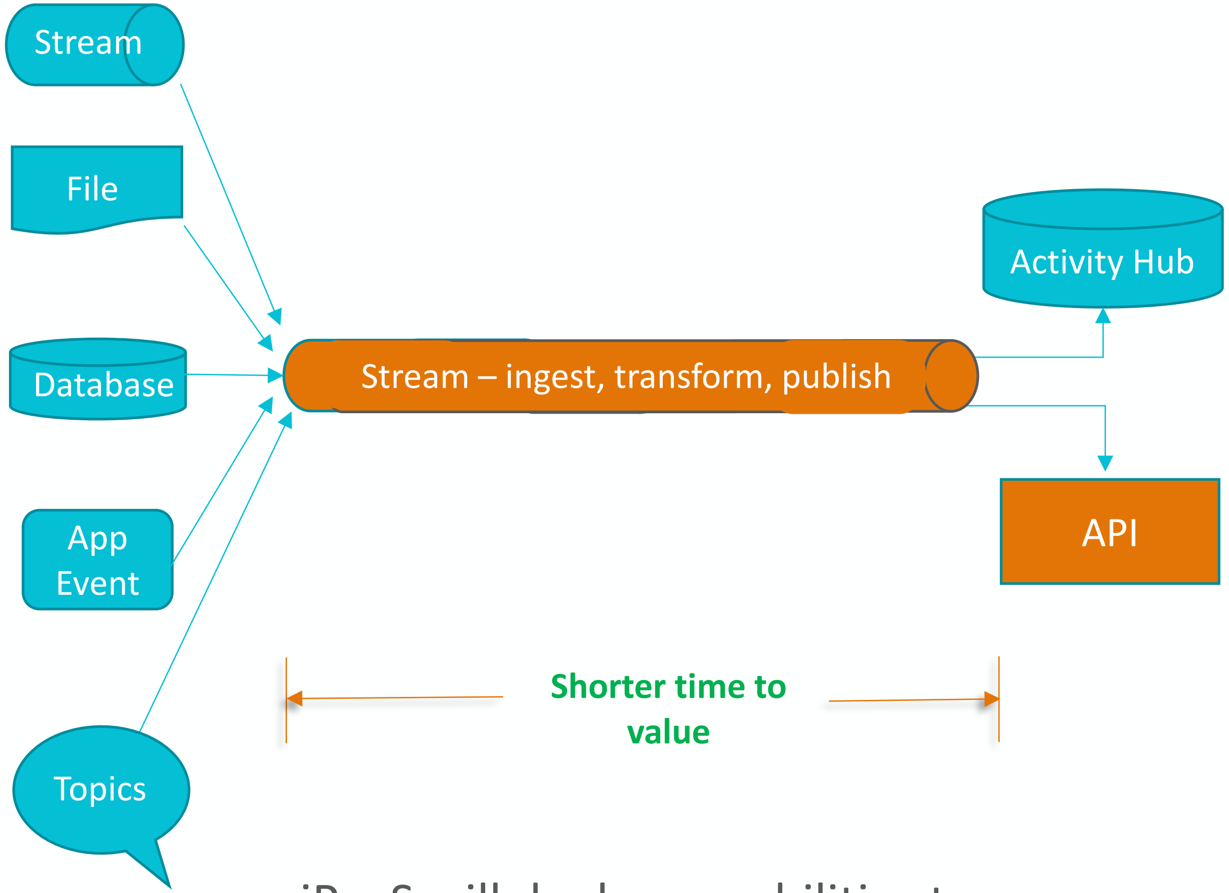
Where are we now and the direction we're heading

Now



iPaaS (and legacy integration) typically works with **data at rest**

Future



iPaaS will deploy capabilities to process **data in-motion** via stream integration

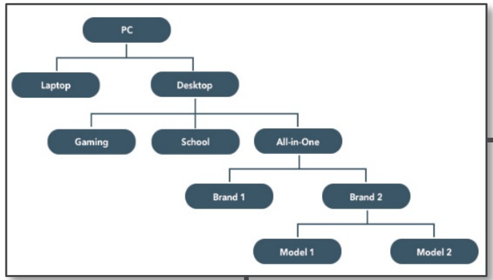
Activity Hub Curated View Design Concepts

<u>Source system/device</u>

Either:

a. Emit from point of entry, full incremental
b. Simulate incremental from DB

<u>Hierarchy manager</u>



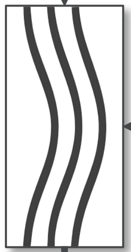
Hierarchy slot attributes

<u>Curated views (CVs)</u>

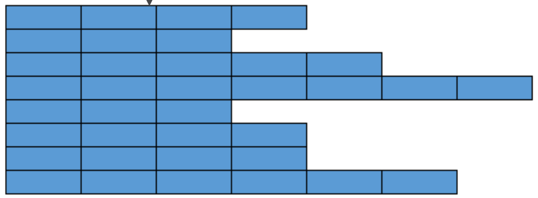
1. Built off of activity records only
2. No base tables
3. CVs are built on top of viewlets
4. CVs can also be built on top of other CVs
5. Viewlet reuse should be high
6. Reuse should be at the highest level
7. CVs eliminate the need for user to do joins
8. CVs are normally materialized
9. Viewlets can also be materialized

<u>iPaaS</u>

a. Simple, parallel streams
b. Minimal hops, steps, merging
c. Save transformation for CVs
d. Easily restartable
e. Save extra data in a bag

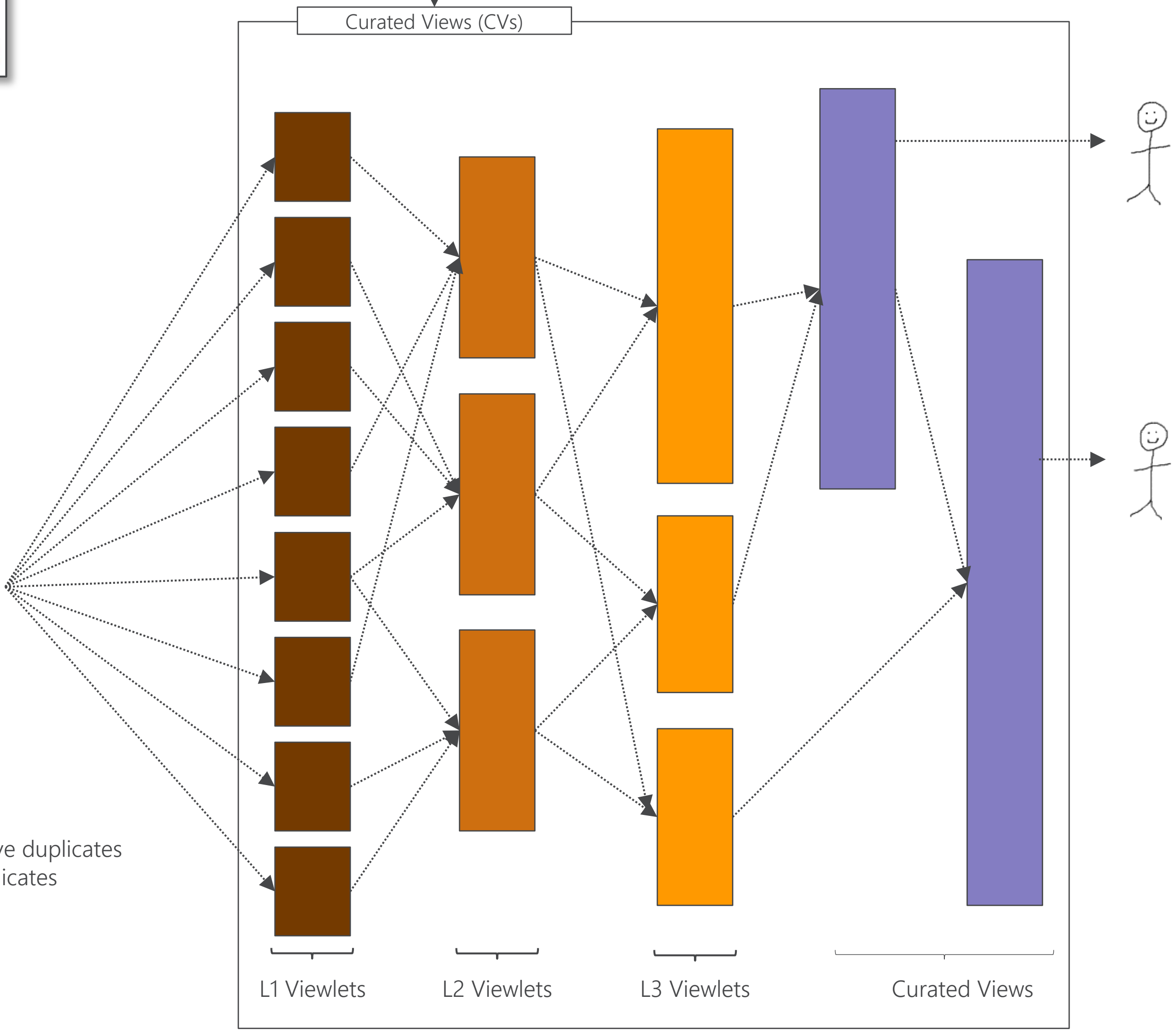


Hierarchy slot ID + [attributes]



<u>Activity table (pile file)</u>

1. Records have different length
2. Record have different fields
3. Records are added in the order they arrive
4. Adds, updates, deletes are different records
5. Records are from idempotent stream and can have duplicates
6. Records have unique identifiers for resolving duplicates
7. An activity table is a replayable log

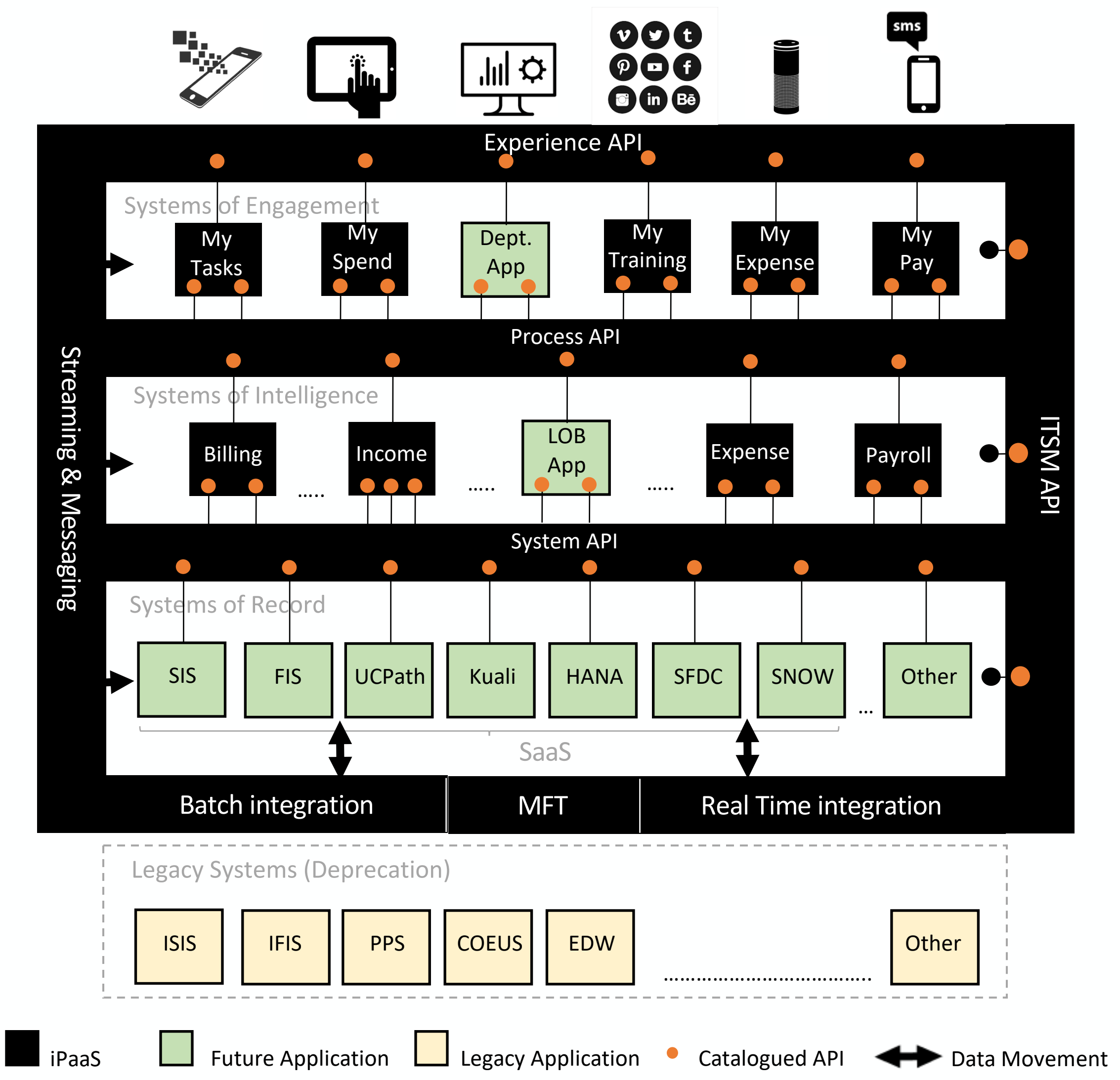


<u>Other notes</u>

- *Use the hierarchical nature of CVs! Do no work too low in the hierarchies!*
- *Merging hierarchy values can be done in-stream or in CV construction*
- *Hierarchy manager will have an activity hub of its own in the Workflow Activity Hub*
- *Keep streams and integration simple, atomic, restartable*
- *CVs need to handle duplicate records, deletions*

VJK: 2/2/2019

What is iPaaS? “Swiss Army Knife” Cloud integration platform



Integration Platform as a Service (iPaaS) is a cloud integration platform that combines application and data integration. It enables development, execution, and governance of integration workflows among on-premise or cloud-based applications as well as traditional and newer data protocols.

iPaaS will provide ESR with uniform methods for the following capabilities via API templates, integration connectors and platforms:

- **Experience API -** API templates to choreograph user experience for a specific device and channel.
- **Process API –** API templates for intelligent, orchestrated business processes.
- **System API –** API templates for access to the full portfolio of core business for master and transaction data.
- **ITSM API** – API templates from COTS solutions that provide IT service management capabilities .e.g. ServiceNow, Jira, etc.
- **Real Time and Batch Integration** connectors for applications that do not support APIs or is scheduled for future deprecation.
- **Streaming** platform for transfer of data at a steady high-speed rate sufficient to support such sensor-based (IoT) applications.
- **Messaging** platform to broker messages through the proper connectors, persistence, authentication, and authorization.
- **Managed file transfer (MFT)** for capabilities to move files from source to target destinations.

iPaaS Platform

Apache Kafka

- Publish and subscribe to streams of records, similar to a message queue or enterprise messaging system
- Store streams of records in a fault-tolerant durable way
- Process streams of records as they occur

Apache Nifi

- A real-time data logistics platform
- Guaranteed Delivery
- Data Buffering w/ Back Pressure
- Prioritized Queuing
- Flow Specific QoS
- Data Provenance
- Configuration based development

WSO2 API Manager

- Publish API Products and Govern the Use of APIs
- Control Access and Enforce Security
- Self service subscription and key management
- Developer Portal to Manage Developer Community
- Manage and Scale API Traffic
- Pluggable and, Extensible

GoAnywhere

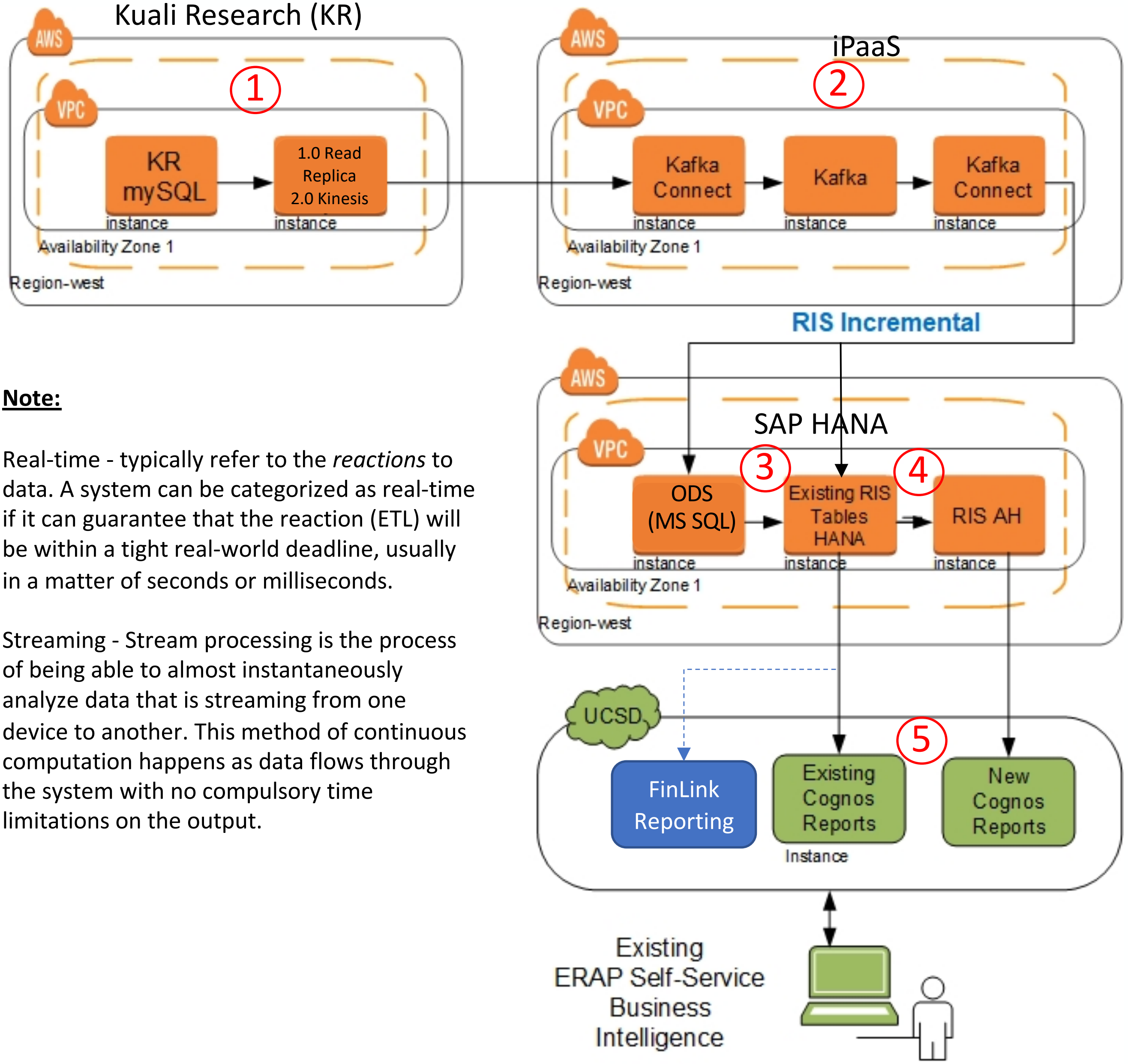
- Managed File Transfer
- Encrypted data at rest and in motion
- Centralized control over all file access
- Translate to/from file formats and databases
- Monitors for FTP, FTPS, SFTP, Azure, Amazon, and LAN

iPaaS Platform

Airflow

- Directed Acyclic Graphs
- Manage scheduling and running jobs and data pipelines
- Programmatic Workflow Management:
- Task Dependency management:
- Manage the allocation of scarce resources
- Monitoring and management interface
- Extendable model and Reusable models
- Retry policy for each task
- Easy interface to interact with logs

Integration Pattern Catalog drives the architecture



Note:

Real-time - typically refer to the *reactions* to data. A system can be categorized as real-time if it can guarantee that the reaction (ETL) will be within a tight real-world deadline, usually in a matter of seconds or milliseconds.

Streaming - Stream processing is the process of being able to almost instantaneously analyze data that is streaming from one device to another. This method of continuous computation happens as data flows through the system with no compulsory time limitations on the output.

The Reporting View provides the future state design that reuses the existing RIS ERAP Reporting & Analytics Services which has a rich self-service BI governance model that will be preserved.

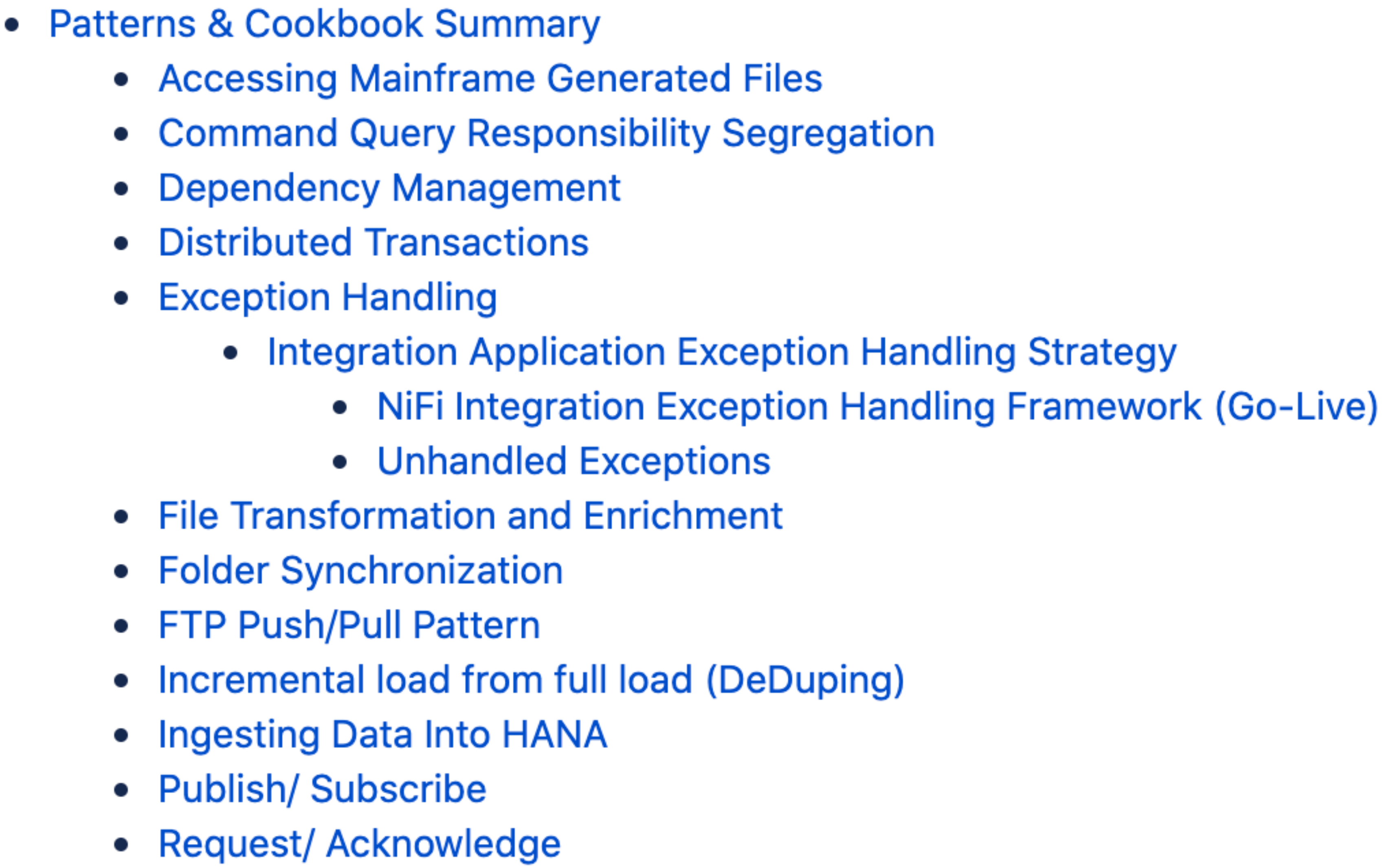
Data Logistics for future state design are as follows:

1. v1.0 - KR will create an AWS RDS Read Replica < 1 sec latency v2.0 – KR will stream at transaction level via Kinesis
2. iPaaS platform will use Kafka connect to collect the mySQL transaction log where Kafka will prepare it (via Kafka Connect) for loading into ODS.
3. SAP HANA platform will store the data in the ODS (MS SQL), and in HANA to recreate the existing RIS tables to satisfy existing reporting requirements *.
4. New reporting requirements will be generated from curated views in the RIS Activity Hub (AH).
5. All existing and new reports will designed, maintained and rendered via existing ERAP solution via Cognos **.
6. Post award data will need to passed to FinLink for existing reporting.

* If transformation is needed to any of the data being loaded into the HANA platform, it is assumed that HANA data transformation tools will be used e.g. Smart Data Integration .

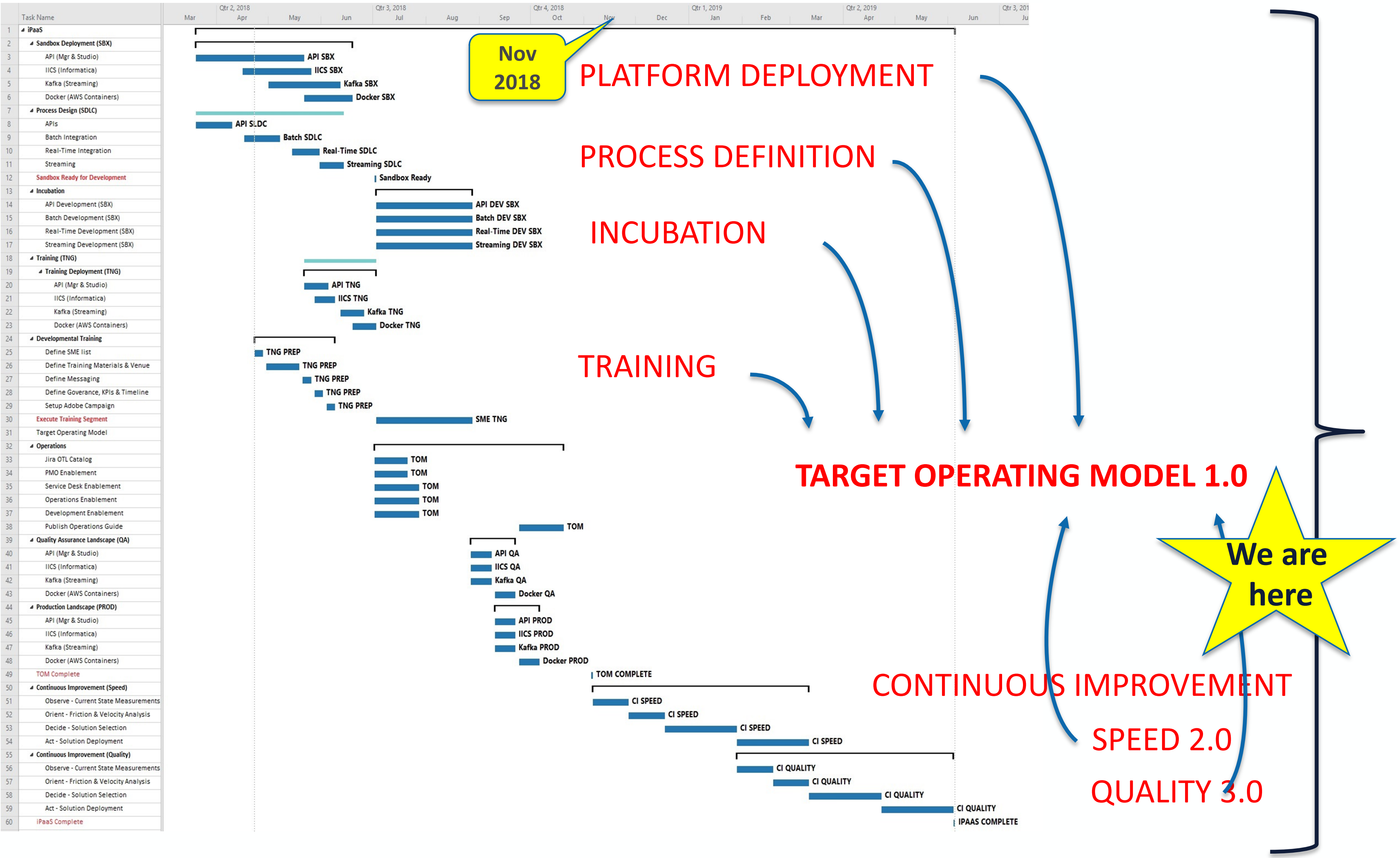
** For more information on ERAP Reporting & Analytics Services visit: https://blink.ucsd.edu/research/data-analysis/erap-reporting-project/index.html

Integration Pattern Catalog drives the architecture



- Patterns & Cookbook Summary
  - Accessing Mainframe Generated Files
  - Command Query Responsibility Segregation
  - Dependency Management
  - Distributed Transactions
  - Exception Handling
    - Integration Application Exception Handling Strategy
      - NiFi Integration Exception Handling Framework (Go-Live)
      - Unhandled Exceptions
  - File Transformation and Enrichment
  - Folder Synchronization
  - FTP Push/Pull Pattern
  - Incremental load from full load (DeDuping)
  - Ingesting Data Into HANA
  - Publish/ Subscribe
  - Request/ Acknowledge

iPaaS timeline for a Target Operating Model (TOM)



	Task Name
1	iPaaS
2	Sandbox Deployment (SBX)
3	API (Mgr & Studio)
4	IICS (Informatica)
5	Kafka (Streaming)
6	Docker (AWS Containers)
7	Process Design (SDLC)
8	APIs
9	Batch Integration
10	Real-Time Integration
11	Streaming
12	Sandbox Ready for Development
13	Incubation
14	API Development (SBX)
15	Batch Development (SBX)
16	Real-Time Development (SBX)
17	Streaming Development (SBX)
18	Training (TNG)
19	Training Deployment (TNG)
20	API (Mgr & Studio)
21	IICS (Informatica)
22	Kafka (Streaming)
23	Docker (AWS Containers)
24	Developmental Training
25	Define SME list
26	Define Training Materials & Venue
27	Define Messaging
28	Define Goverance, KPIs & Timeline
29	Setup Adobe Campaign
30	Execute Training Segment
31	Target Operating Model
32	Operations
33	Jira OTL Catalog
34	PMO Enablement
35	Service Desk Enablement
36	Operations Enablement
37	Development Enablement
38	Publish Operations Guide
39	Quality Assurance Landscape (QA)
40	API (Mgr & Studio)
41	IICS (Informatica)
42	Kafka (Streaming)
43	Docker (AWS Containers)
44	Production Landscape (PROD)
45	API (Mgr & Studio)
46	IICS (Informatica)
47	Kafka (Streaming)
48	Docker (AWS Containers)
49	TOM Complete
50	Continuous Improvement (Speed)
51	Observe - Current State Measurements
52	Orient - Friction & Velocity Analysis
53	Decide - Solution Selection
54	Act - Solution Deployment
55	Continuous Improvement (Quality)
56	Observe - Current State Measurements
57	Orient - Friction & Velocity Analysis
58	Decide - Solution Selection
59	Act - Solution Deployment
60	iPaaS Complete

Nov 2018

PLATFORM DEPLOYMENT

PROCESS DEFINITION

INCUBATION

TRAINING

TARGET OPERATING MODEL 1.0

We are here

CONTINUOUS IMPROVEMENT

SPEED 2.0

QUALITY 3.0

Implications

Without a target operating model improving the capabilities over time will be difficult.

Currently ~15% variance in timeline and ~5% defect rate

UC San Diego
INFORMATION TECHNOLOGY SERVICES

Service Maturation

Observability – too many tools = swivel chair operations, must start journey toward AIOps

Reliability – platform deployed and scaled to 100’s of workloads in less than a year, maturation is a must

Cost optimization – cloud is expensive, must move to just-in-time computing model driven by duty cycle

Service delivery - UCSD is a service provider RACI, Service Descriptions, SLA, OLAs, etc.

Center of Excellence – starting Integration community of practice