GridGain

In-Memory Computing Best Practices

Developing New Applications, APIs and Analytics

Rob Meyer

Outbound Product Management

January 24, 2019

In-Memory Computing Best Practices Series

1. Build an in-memory computing foundation and roadmap
2. Add speed and scale to existing applications
3. Develop new applications, APIs and analytics
4. Make the bridge to the cloud
5. Implement HTAP and real-time data management
6. Become event- and stream-driven
7. Think different with machine and deep learning

GridGain

Best Practice 3.1 (1.1)

Design Around the Customer Experience

“40 percent of all technology spending will go toward digital transformations, with enterprises spending in excess of $2 trillion in 2019”

Goal of Digital Business: Improve the Customer Experience

HARVARD BUSINESS SCHOOL

Leading digital companies generate **better gross margins**, **better earnings** and **better net income** than organizations in the bottom quarter of digital adopters.

Performance Metric	Digital Laggards *(Bottom 25% of enterprises)*	Digital Leaders *(Top 25% of enterprises)*
Three Year Average Gross Margin[1]	37%	55%
Three Year Average Earnings before Taxes	11%	16%
Three Year Average Net Income	7%	11%

GridGain

Goal of Digital Business: Improve the Customer Experience

“We want to be a tech company with a banking license”

Ralph Hamers, CEO

ING

GridGain

Lieven Merckx

IT Achitect at ING Belgium

Share this page

- Civik Engineer ICT & Electronics
- 32 years ING IT Infrastructure and System Development

https://www.imcsummit.org/2017/eu/users/lieven-merckx

Sessions:

Panel Discussion: The Future of In-Memory Computing

☐ **Wednesday 21st, 9:50 -**

-

Journey into the Use of IMDG to Strengthen the Electronic Banking Offering of ING. BE

☐ **Tuesday 20th, 14:35 - Matterhorn 2**

The presentation will explain how GridGain IMDG is considered fit for purpose to increase the scalability and availability of the electronic banking offerings. The goal is to share experiences on how this technology fits into the technical landscape and which integration patterns are being tested and deployed. Target audience engineers and...

Our strategy

Customers are increasingly digital and bank with us more and more through mobile devices. Their needs and expectations are the same, all over the world, and they expect us to adopt new technology as fast as companies in other sectors. In order to continue to lead in digital banking, we need to offer a better customer experience, that's instant, personal, frictionless and relevant.

5

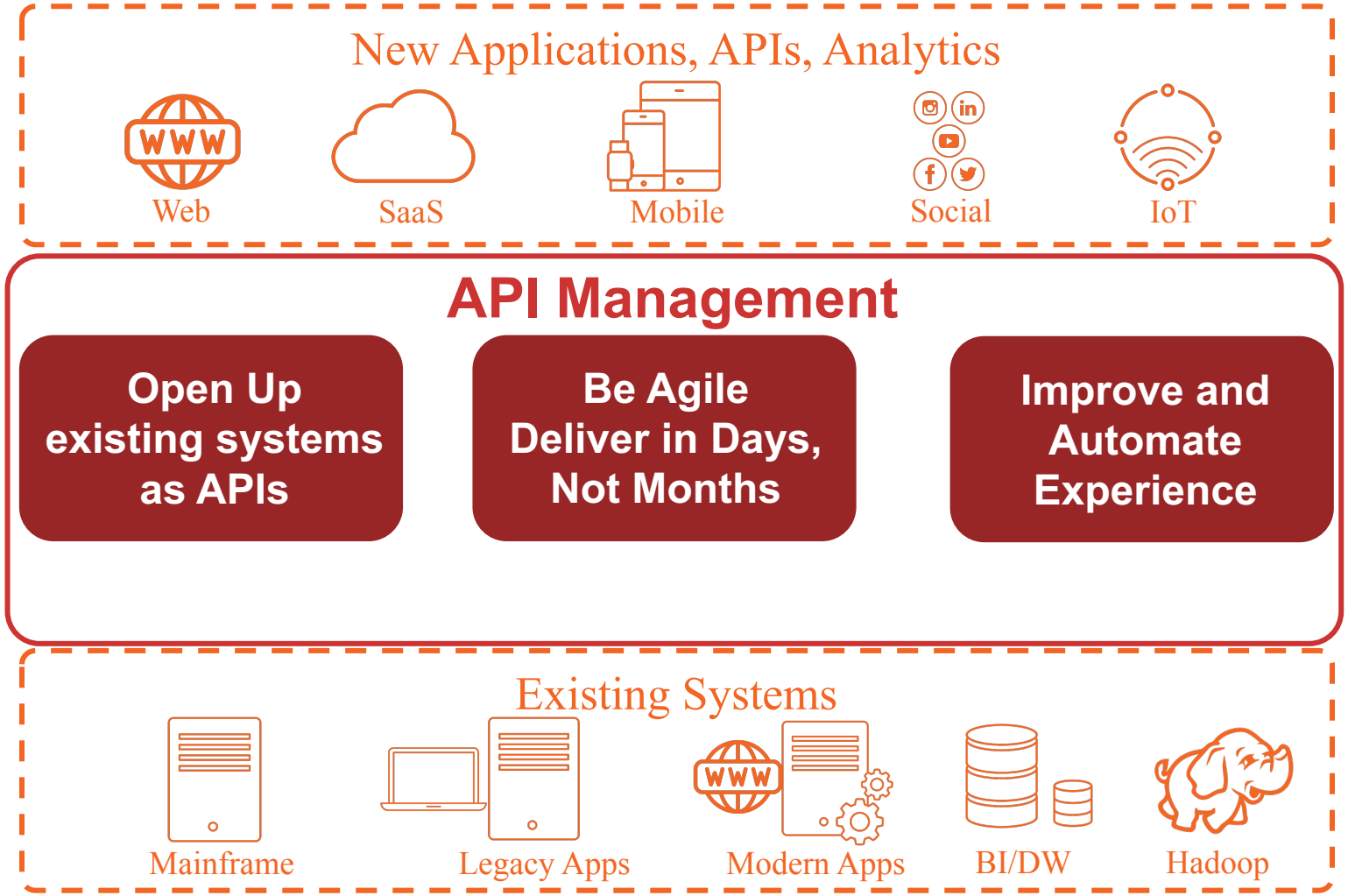
Best Practice 3.1: Design Around the Customer Experience

- People: Identify the right (additional) people and responsibilities across your customer experience and digital business initiatives
  - Chief Customer Officer
  - Process Architect
  - API Product Manager
  - Chief Data Officer
- Process: Design processes and APIs around the customer
- Technology: Design a process-centric API architecture
  - Layer your APIs
    - Public (outside-in contracts) and private (microservices)
    - Process (customer experience) and data (abstraction) services
  - Design (extensible pipelines) for real-time analytics and automation

Best Practice 3.2 (1.2)
Build a New Real-time Layer on Top of Existing Systems

The Principles of API Management

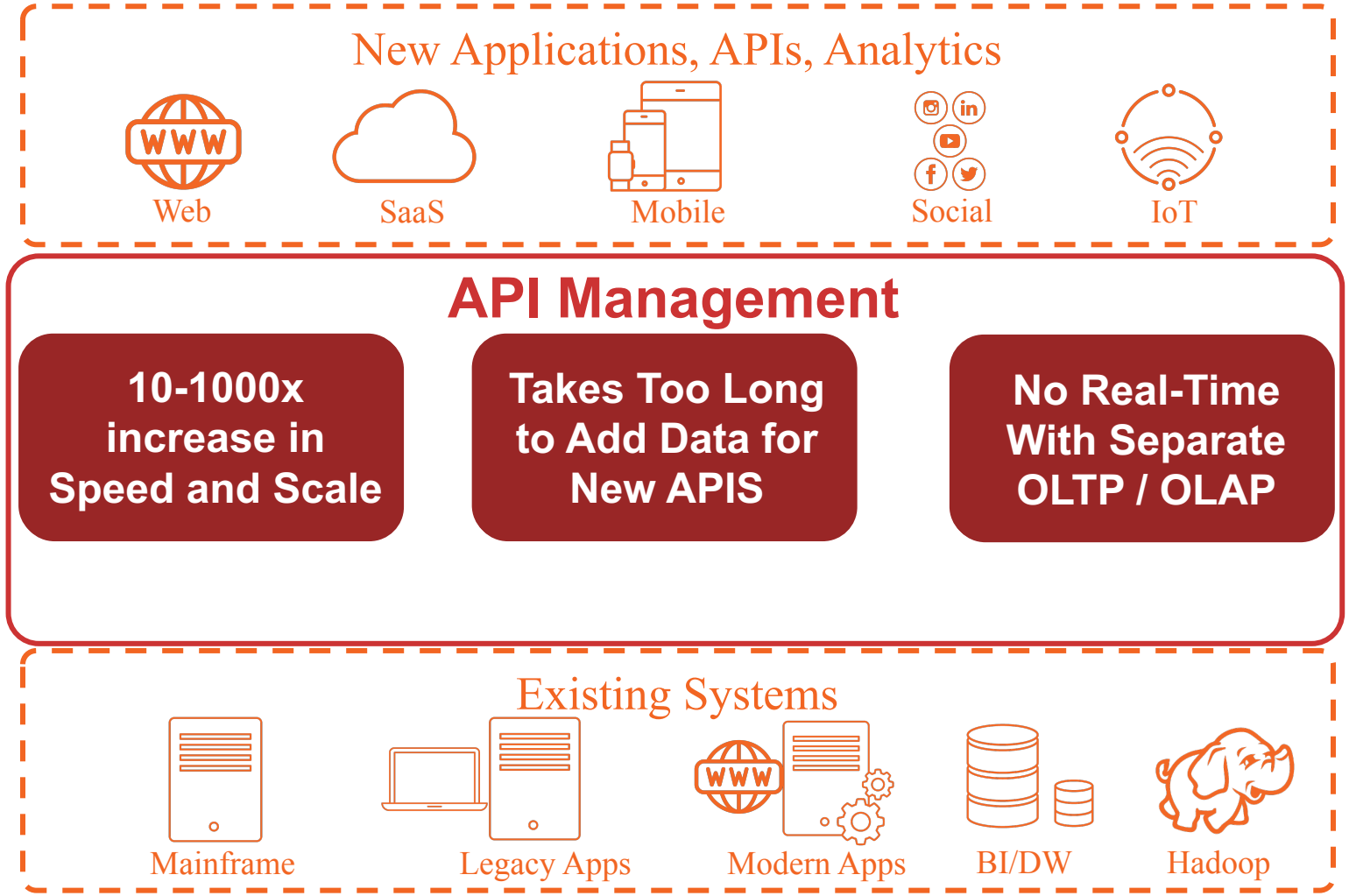
How to Become API First



GridGain

The Speed and Scale Challenges with API Management

Existing Systems Cannot Deliver the Speed and Scale that APIs Need



GridGain

The Speed and Scale Challenges with API Management

Existing Systems Cannot Deliver the Speed and Scale that APIs Need

10-1000x increase in Speed and Scale

- API throttling limits /restricts API usage
- Messaging flow control protects existing systems, violates SLAs
- Database scale-up too expensive, limited

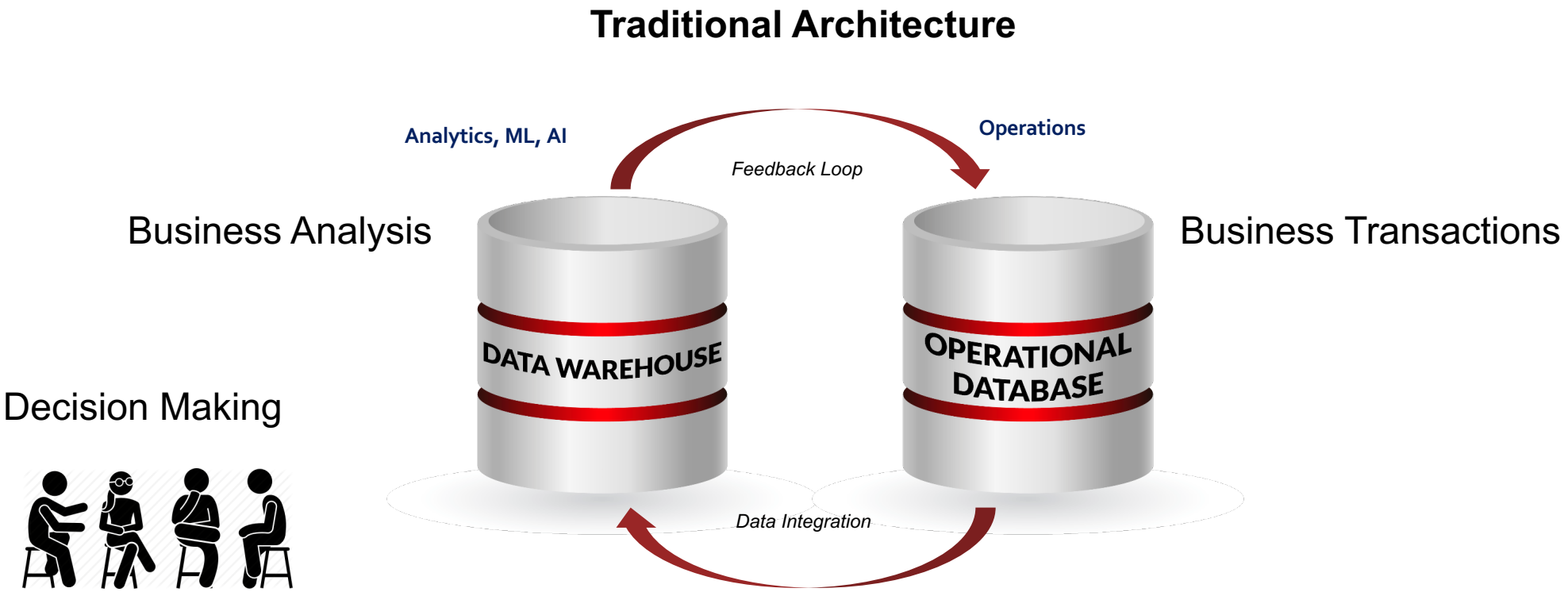
Takes Too Long to Add Data for New APIS

- Access limited by old firewall-based security
- Not allowed to (over)load OLTP systems
- Federated queries are slow
- Deployment too long for old apps

No Real-Time With Separate OLTP / OLAP

- OLAP is separate to protect OLTP systems
- Unable to get real-time access to up-to-date information during interaction
- Unable to run analytics fast enough

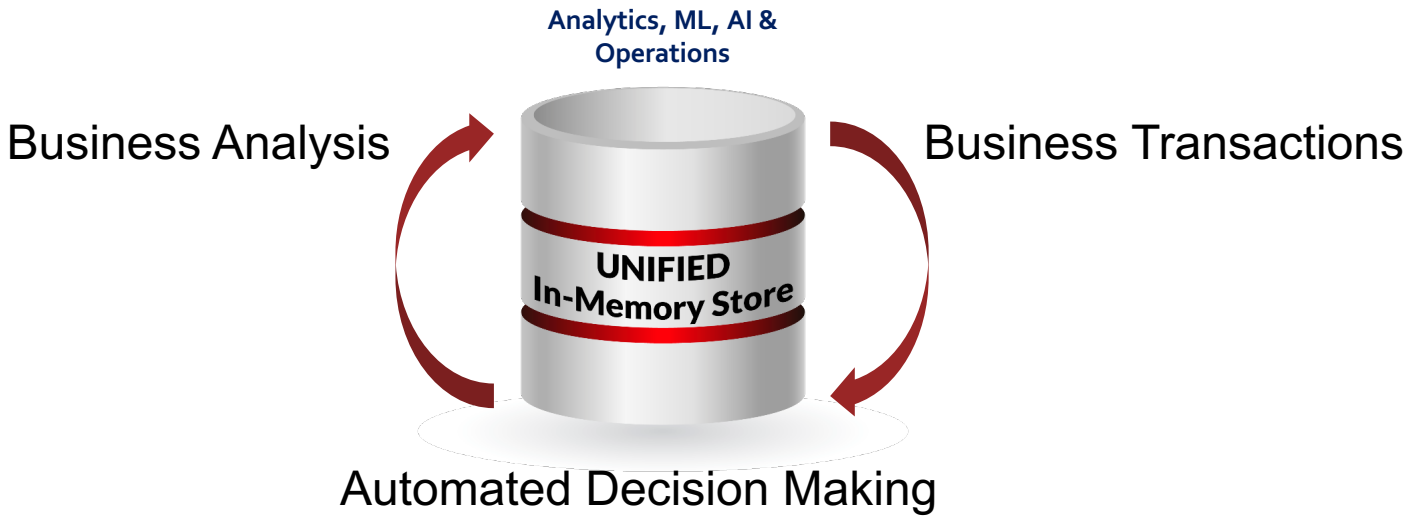
Traditional IT Architectures Cannot Meet the Needs of the Digital Enterprise



In-Memory Computing Platforms will Power the Digital Enterprise

"IMC-enabled HTAP can have a transformational impact on the business."
— Gartner 2/17

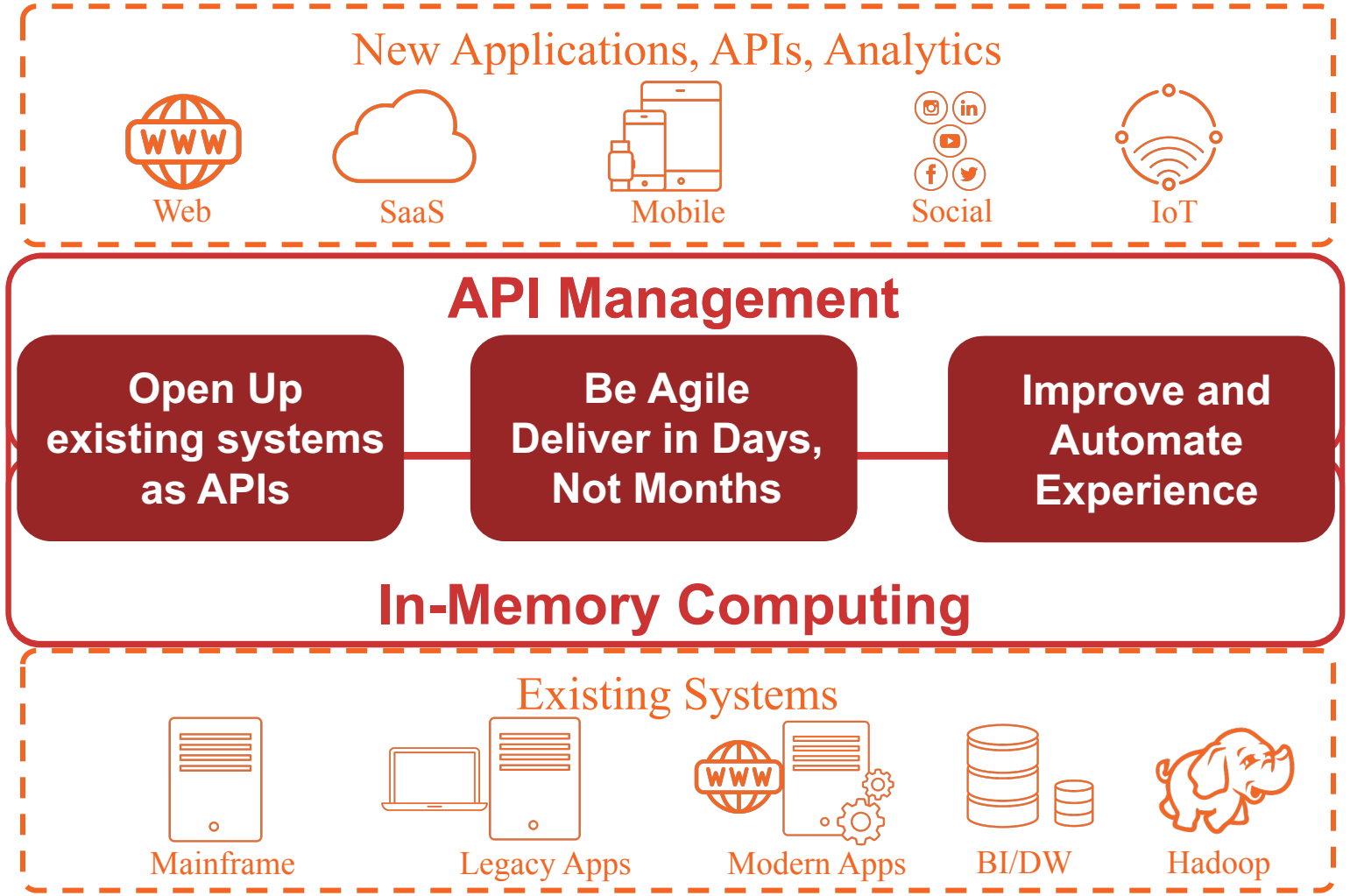
Unified IMC HTAP Architecture



IMC-enabled HTAP enables situational awareness on live operational data as opposed to after-the-fact analysis on stale data

In-Memory Computing - The Real-Time Data Foundation

For API Management, Hybrid Transactional/Analytical Processing (HTAP), and Machine / Deep learning



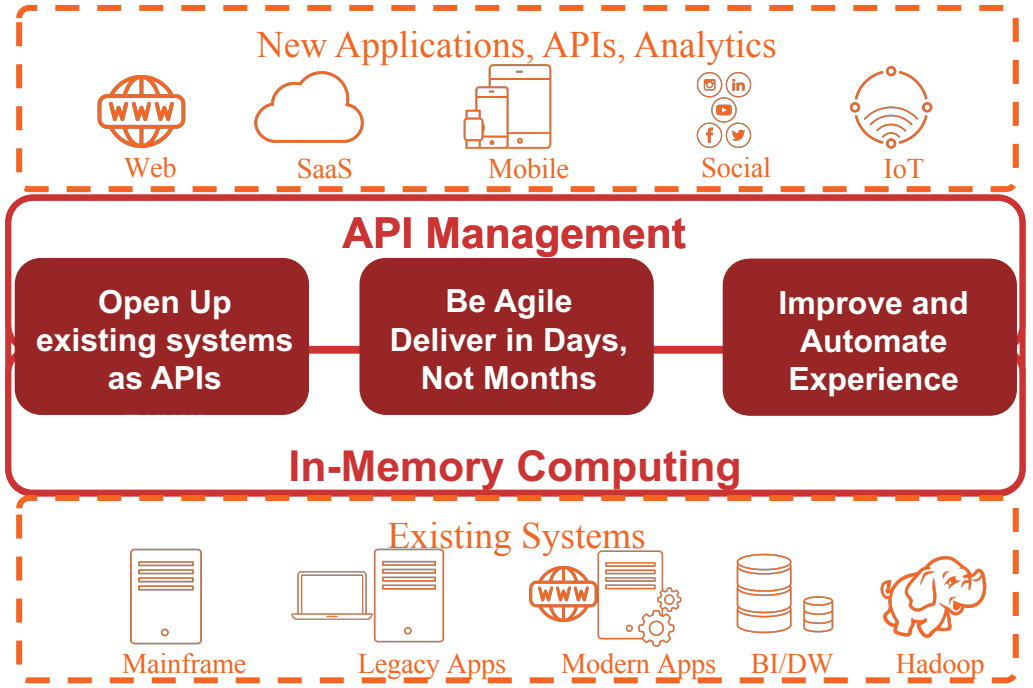
GridGain

In-Memory Computing - The Real-Time Data Foundation

For API Management, Hybrid Transactional/Analytical Processing (HTAP), and Machine / Deep learning

A New Data Layer

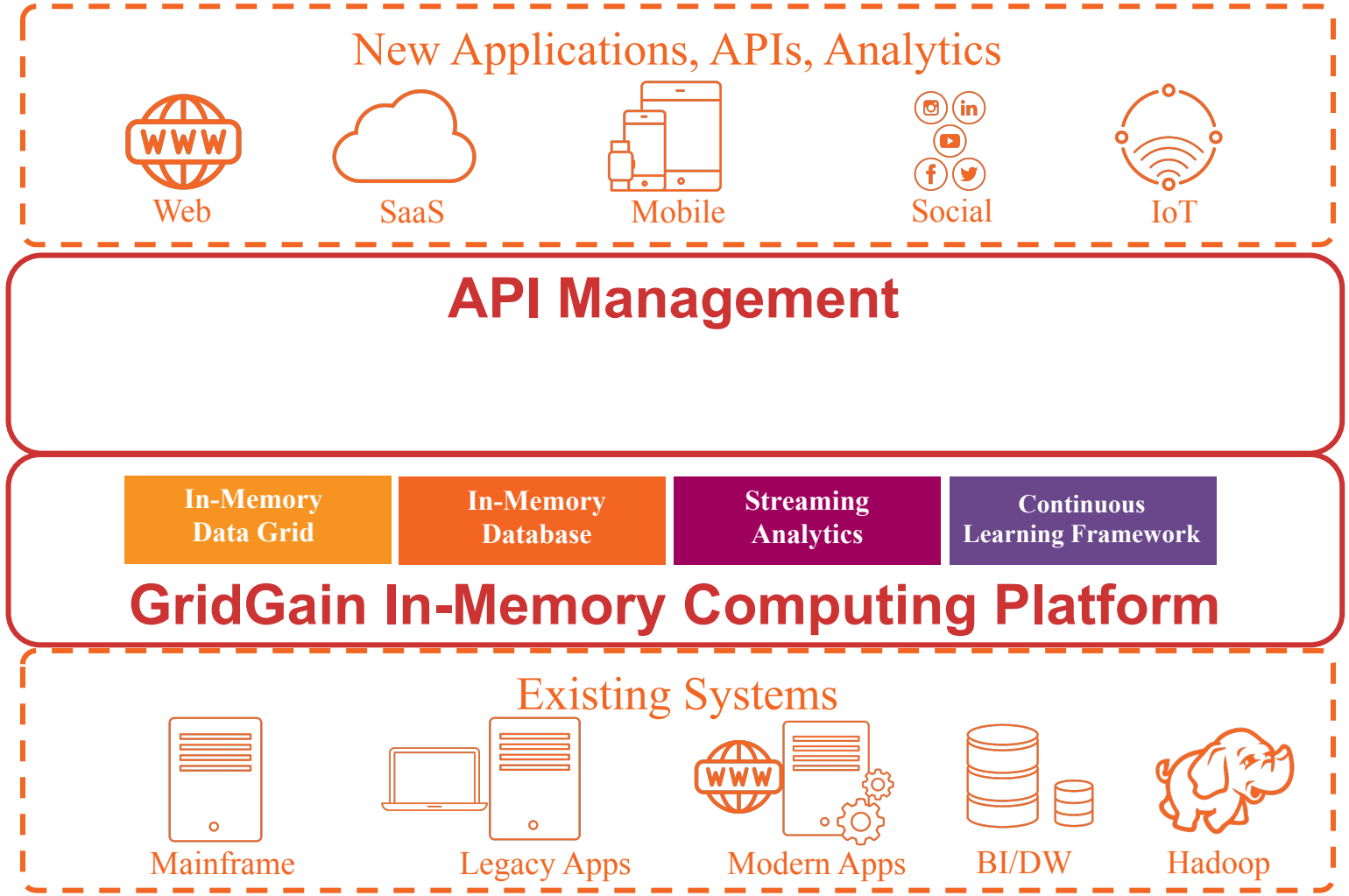
- **For speed:** collocates data in memory with each API
- **For scale:** scales horizontally, elastically with each API
- **For agility:** allows data to be collocated with each API without putting a load on existing apps
- **To improve and automate the experience:** enables real-time analytics and ML/DL by moving any compute to the data



GridGain

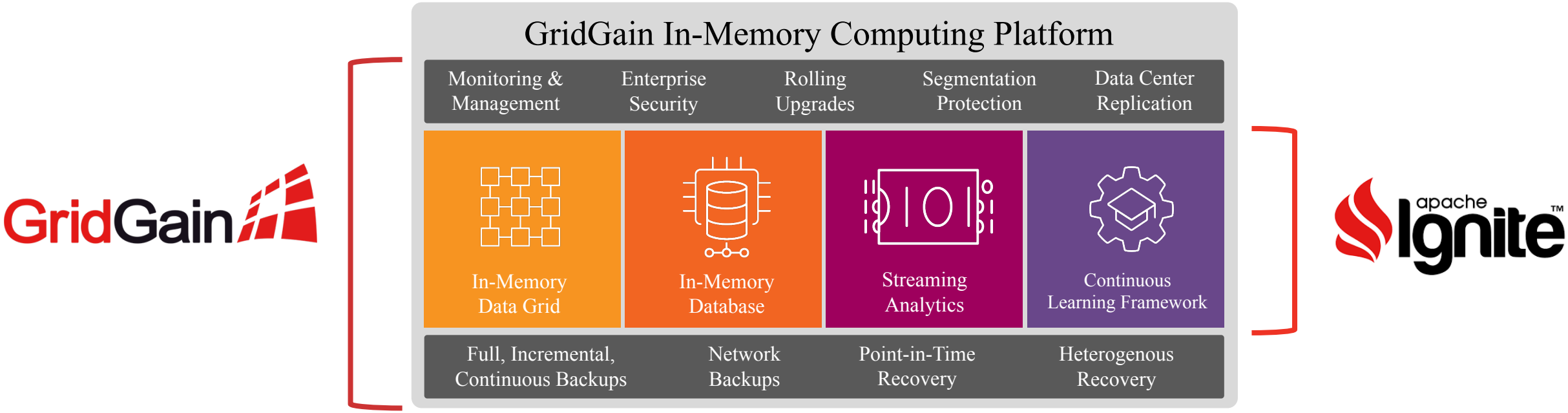
The GridGain In-Memory Computing Platform

For API Management, Hybrid Transactional/Analytical Processing (HTAP), and Machine / Deep learning



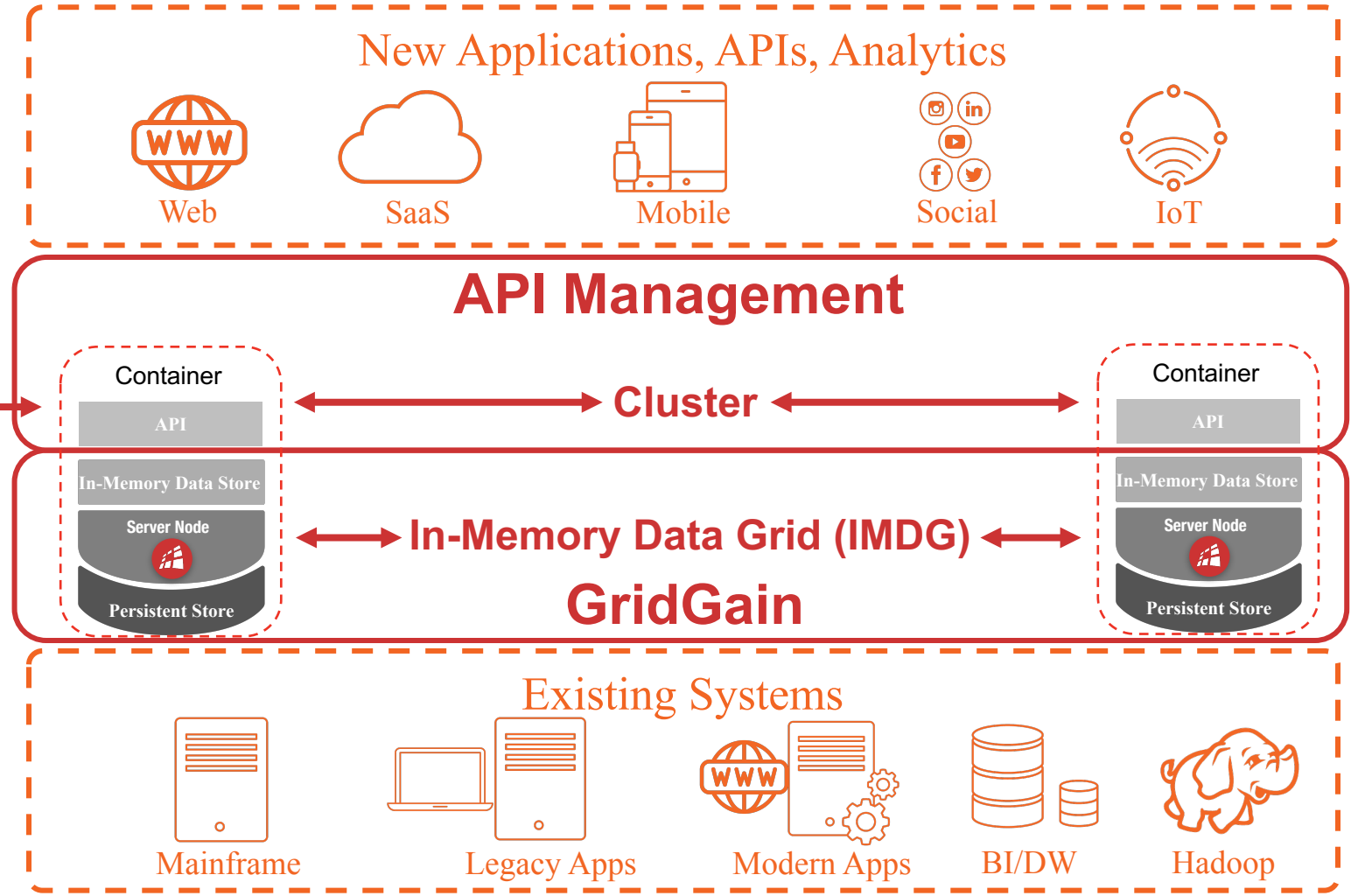
GridGain

GridGain and Apache Ignite Compared



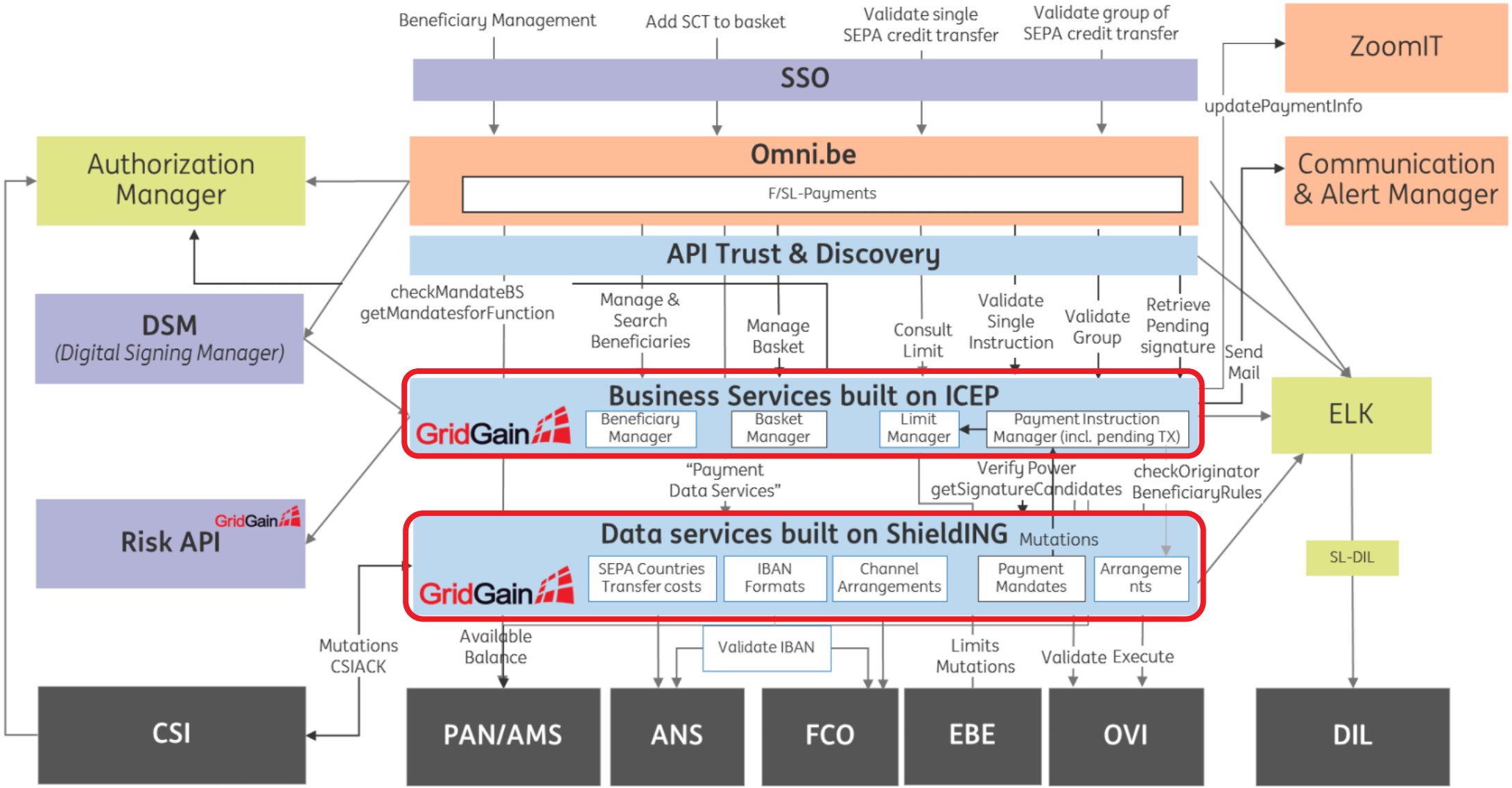
How GridGain is Deployed for API Management

Docker Container (Kubernetes)



GridGain

How does everything fit together?



13

Do The API Math
In-Memory Storage, Horizontal Scale and HTAP are Inevitable

API Math

- Speed: Max data service latency = Max end-to-end responsiveness time (<1 sec)
  - client app processing + WAN hop
  - Max SSO time + LAN hop
  - Max security policy time + LAN hop
  - Max business service time + hop
- Storage scale: Total storage = current data * growth rate * 5 years + new data (year 1) + new data (year 2) + ... New data (year 5)
- Query scale: existing queries * growth rate over 5 years + new queries over 5 years
- Network (compute) latency = total amount of data needed per second / min network bandwidth

HTAP is inevitable with Big Data and Small Pipes. The question is when …

- A 10GigE network can only move 1GB/sec fully loaded …
- There are terabytes of data about customer interactions/behavior …
- Design core services for collocated computing/MPP
  - EITHER move the node to the code …
  - OR move the code to the node

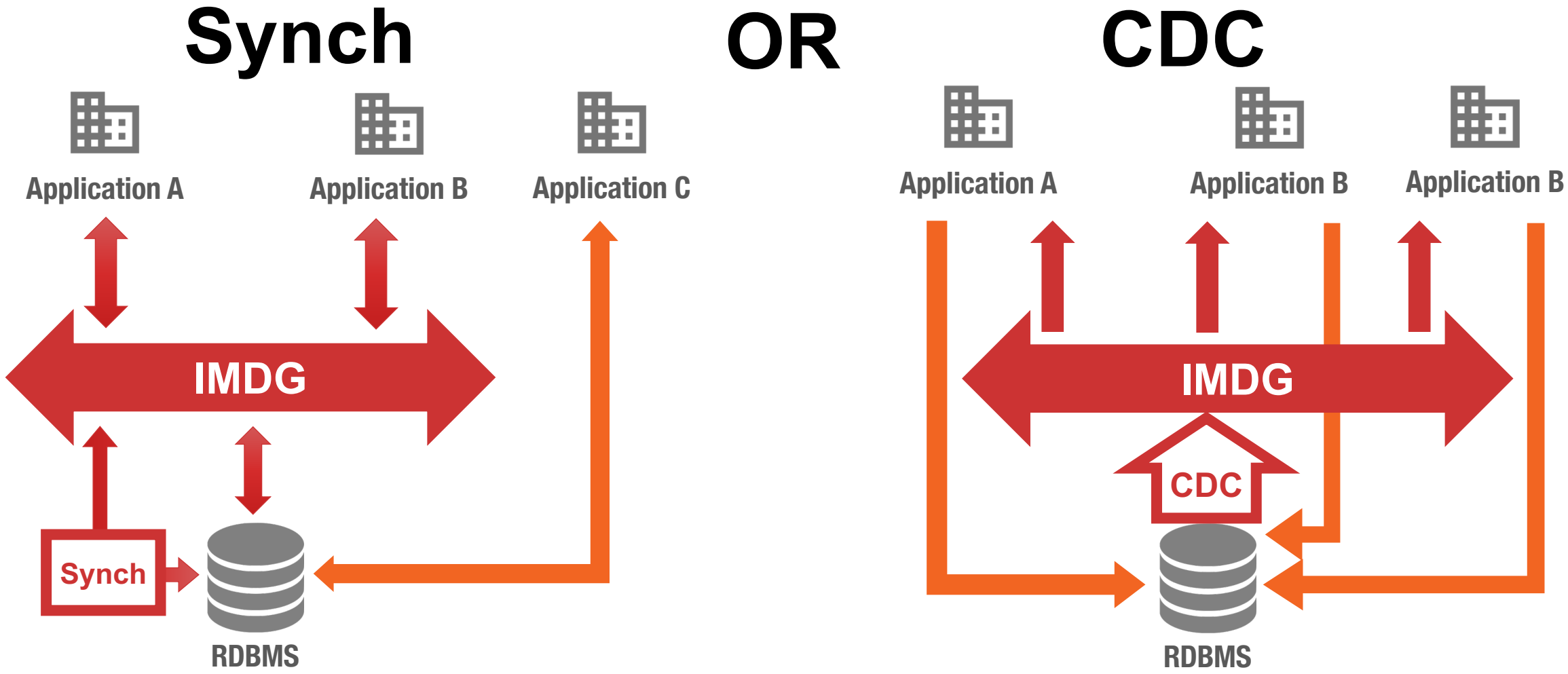
Design for HTAP now

Reminder:
Best Practice 2.2

Choose the Best Integration Architecture between real-time and non-real time apps

Keep The New (Real-Time) and Existing (Batch) in Synch

Build a Bi-Directional Integration Layer



GridGain

Best Practice 3.2: Build a New Real-Time Layer

- Do the API Math
  - Speed
  - Query/transaction scale
  - Storage scale
  - Network scale
- Build a real-time API layer on in-memory computing designed for HTAP
  - Collocate data in-memory with each API as the real-time system of record
  - Make sure you can scale elastically
- Integrate between the new real-time API layer and existing systems
  - Choose between real-time synch or CDC (asynch)
  - Implement bi-directional integration, including updates from existing to new

Best Practice 3.3 (1.3)

Real-Time Digital Business Requires Many Tools

Build Real-Time Data, Events, Tasks and Processes …

One Competency is Never Enough for Transformation

ML / DL

API Management

Business-to-Business (B2B)

Streaming Analytics (BAM 2.0)

Data Integration
Data Quality
Master Data Management

Business Process Management (BPM)

SOA/EAI Messaging

New Omnichannel Experience

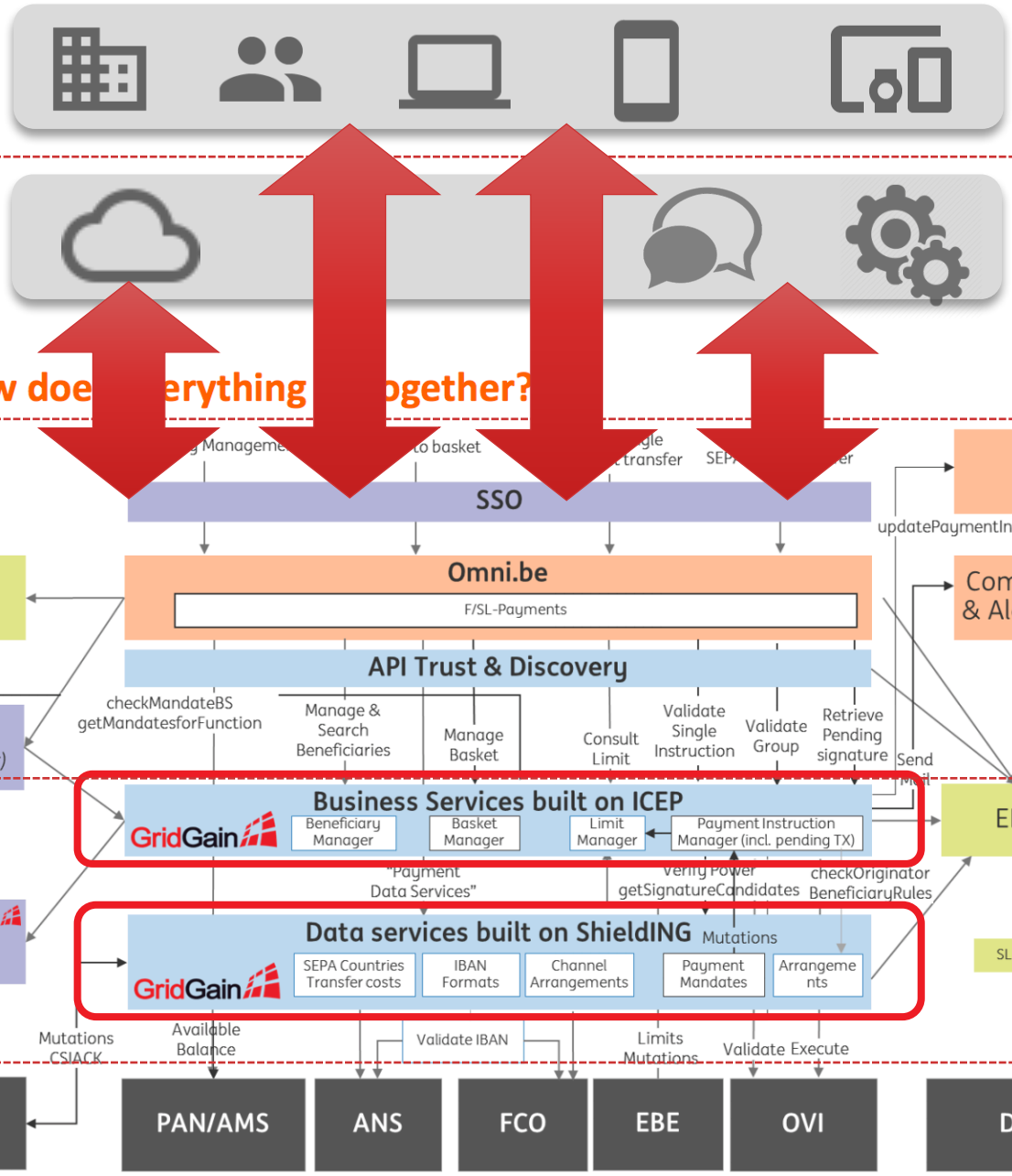
New Technologies and Data (cloud, social/streaming, ML/DL)

Security, Identity Management

Business Services (APIs)

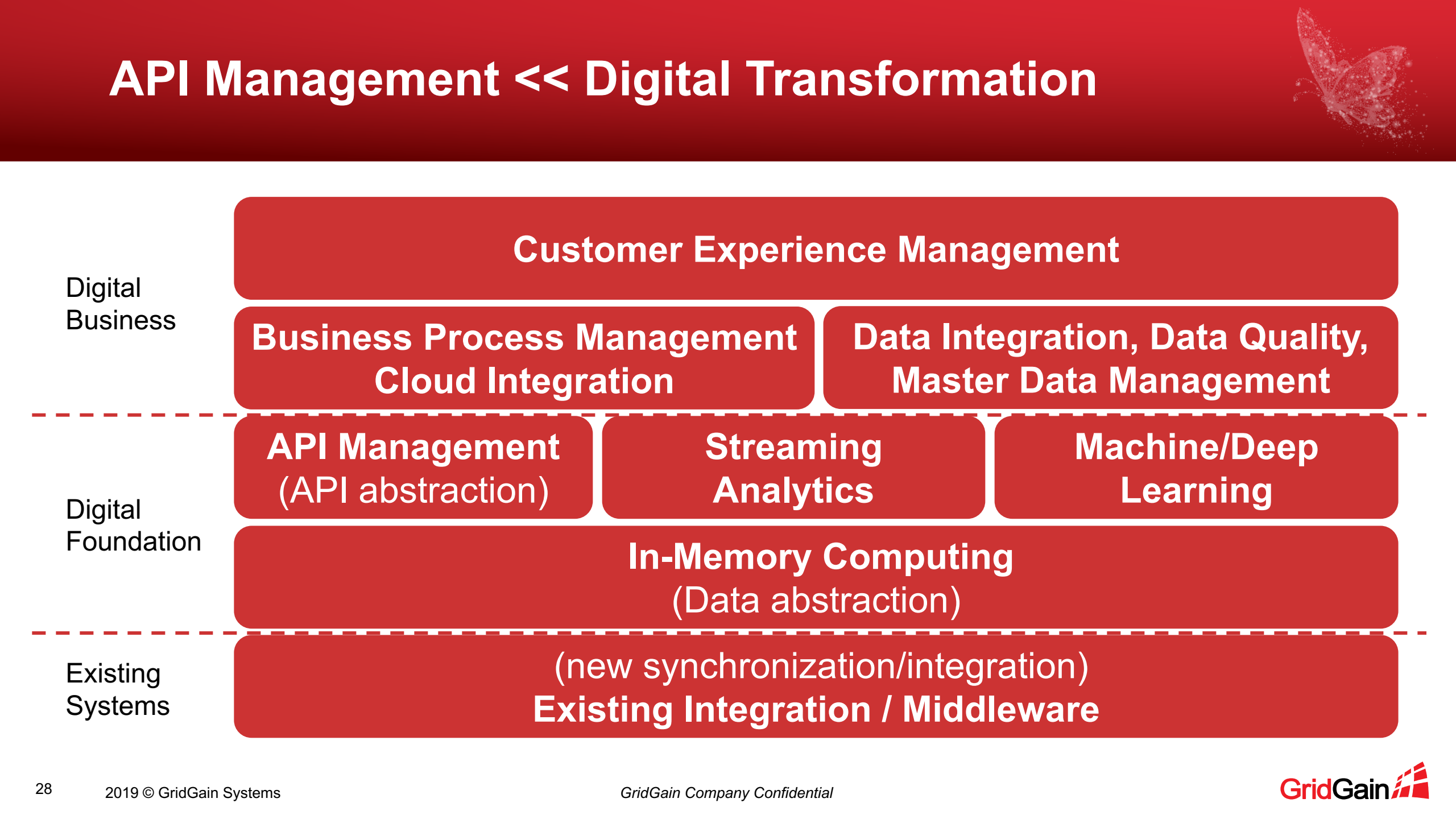
Data Services (APIs)

Existing channels

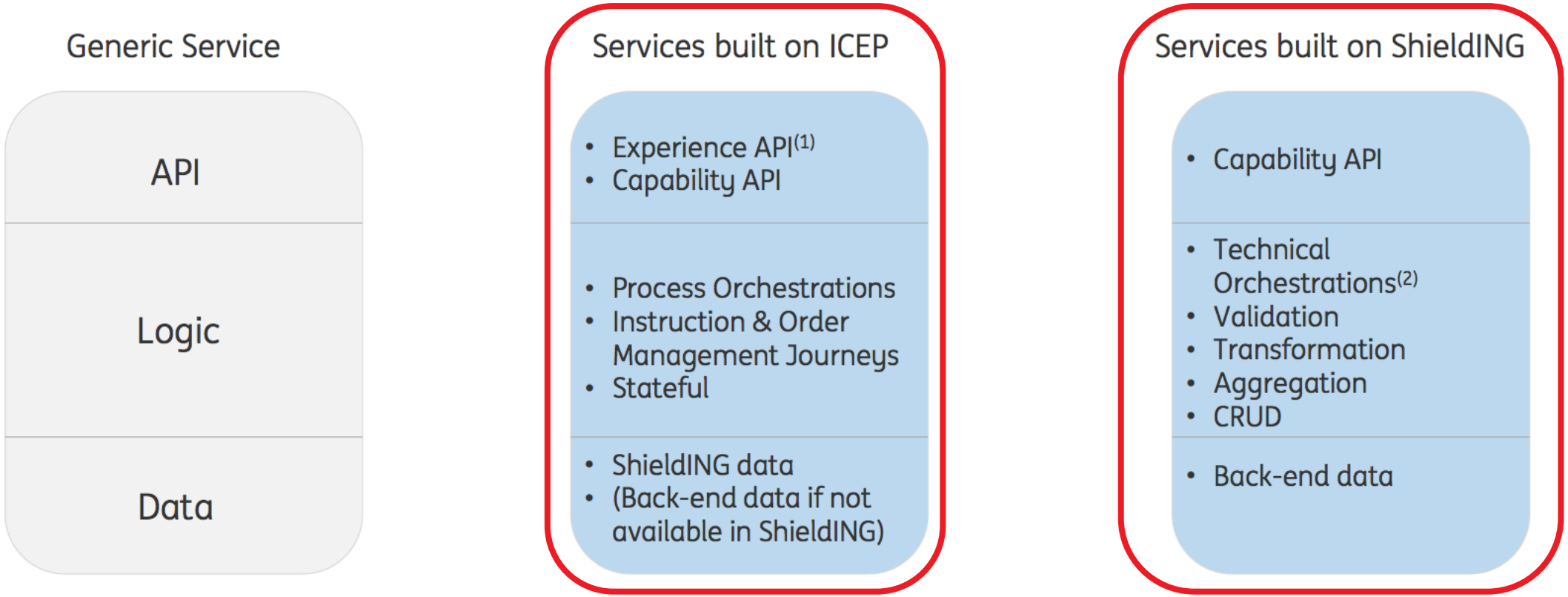


13

API Management << Digital Transformation



Key concepts & definitions – Difference between ICEP and ShieldING Services

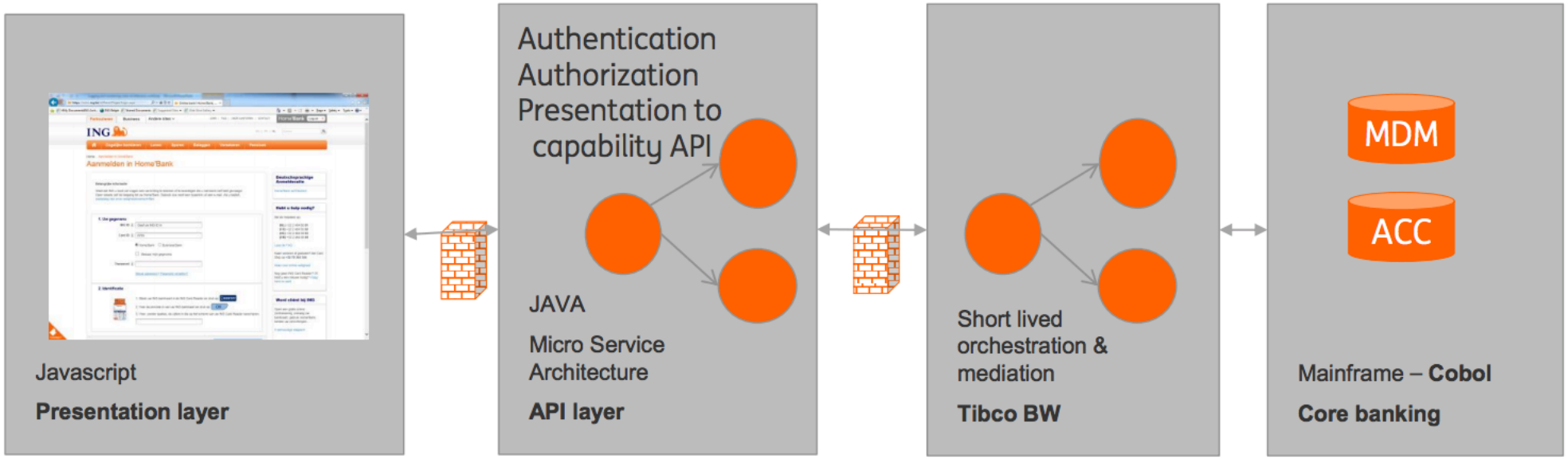


(1) Experience API provide the operations directly supporting the Distribution Application Dialogs and typically correspond to stateless Business Journeys. Currently these API are developed as NSP SL components. These API will gradually be migrated to API on ICEP and ShieldING. See Appendix for description of TPA API.

(2) Technical orchestrations typically involve multiple service calls to retrieve the requested information (where the output of one call is used as input for the other)

14

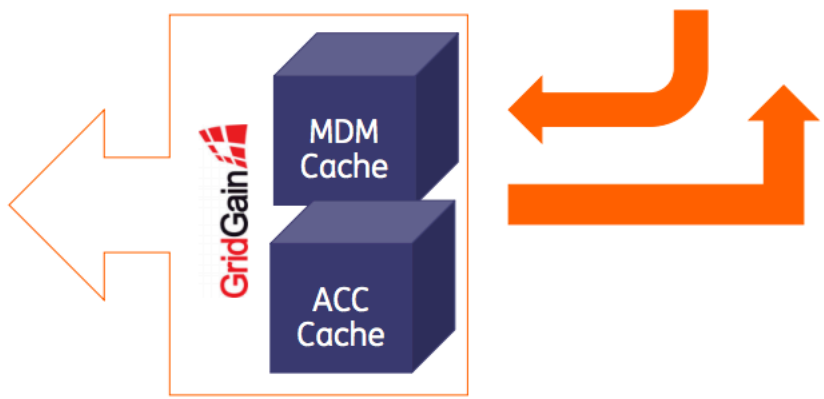
Shield-ING : Data services approach for mainframe offload



AS-IS

Benefits

- Increase availability of the essential data
- Decouple Mainframe availability and performance
- Increase performance
- Data transformation
- Linear horizontal scalable



TO-BE

ING

Best Practice 3.3 (1.3): Build Real-time Data, Events, Tasks and Processes

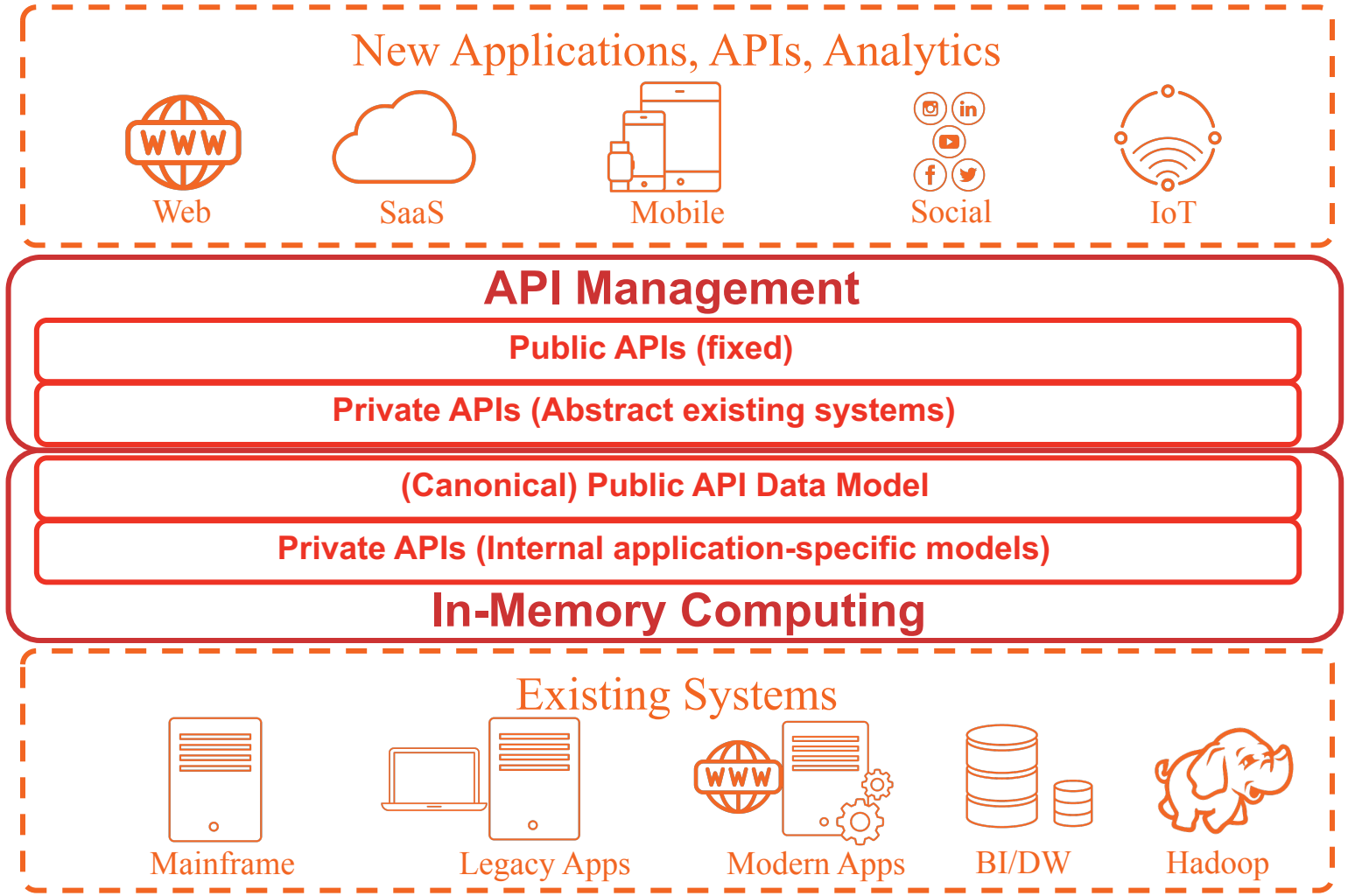
- Define your overall real-time architecture up front
- Design for extensibility
- Choose when to adopt technologies based on your roadmap and ROI
  1. Data Integration (DI), Quality (DQ), Master Data Management (MDM)
  2. New data ingestion (real-time persistence)
  3. Streaming analytics
  4. Machine / deep learning
- Leverage newer streaming technologies together: KISS
  - Kafka: Events/streaming
  - Ignite: Data, collocated processing
  - Spark: Stream processing
  - Spring (Boot/Data): Tasks, APIs, processing

Best Practice 3.4 (2.3)

Free Your Data! Now!

API Management and In-Memory Data Layer

How to Abstract to Support Agility and Change



GridGain

When to Move Data To In-Memory Computing To Open Up Your Data

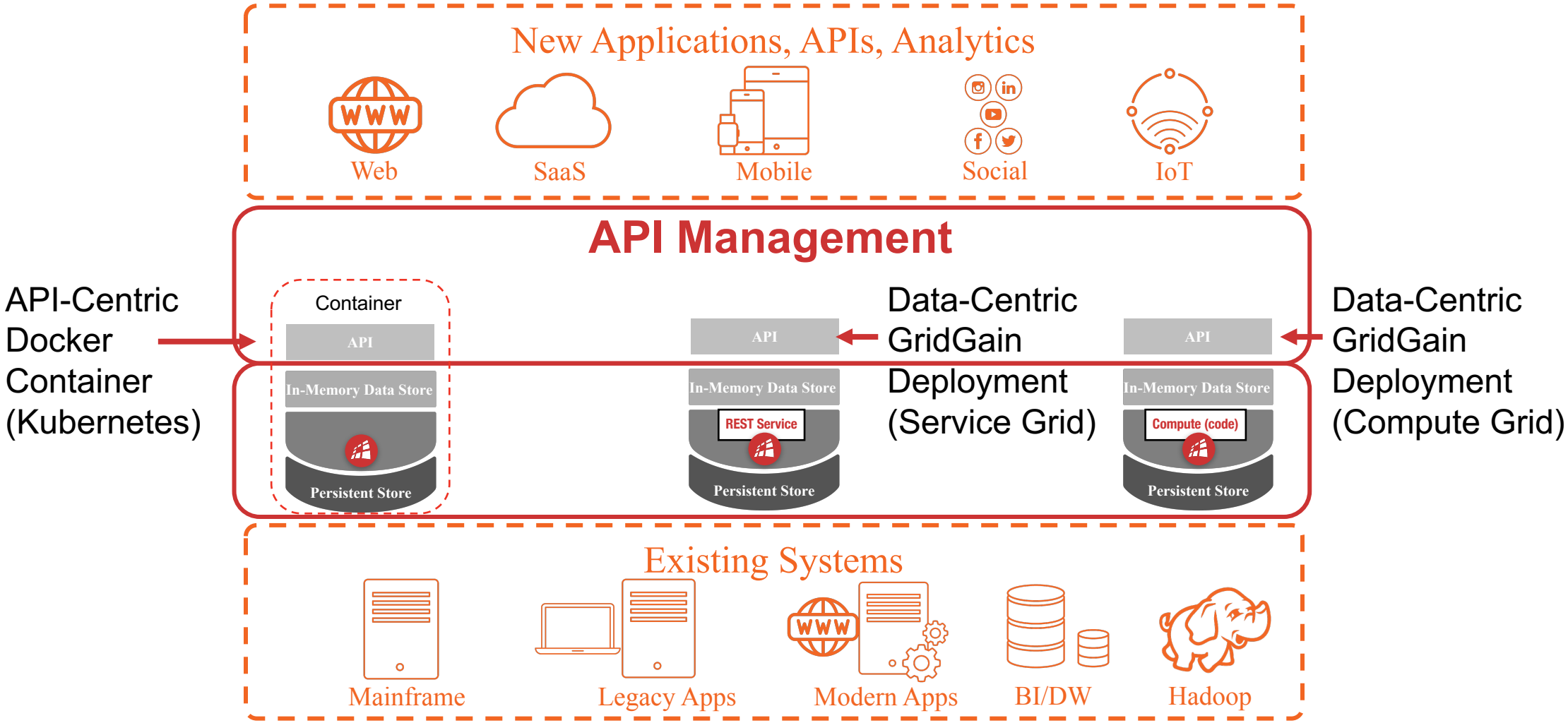
- For new APIs
  - Enables speed and scale without impacting existing apps
- For HTAP
  - Enables new analytics to be performed against data
- For future projects
  - Make sure to plan ahead to support future projects
  - May provide a better ROI
- To simplify integration

Best Practice 3.4 (2.3): Free Your Data! Now!

- Decide when existing data needs to be unlocked for new APIs, HTAP, …
  - Move it to the common in-memory data layer (IMDG) for existing applications either when existing data is needed for a project OR speed/scale is needed
  - Decide if it's better to implement above existing apps or below
  - Persist in GridGain when transactional volumes need to scale
- Create data layers in line with your API architecture
  - Private APIs with application-specific knowledge of data
  - Public APIs that use a common (canonical) API data model

Best Practice 3.5 (2.6)
Design with HTAP in Mind

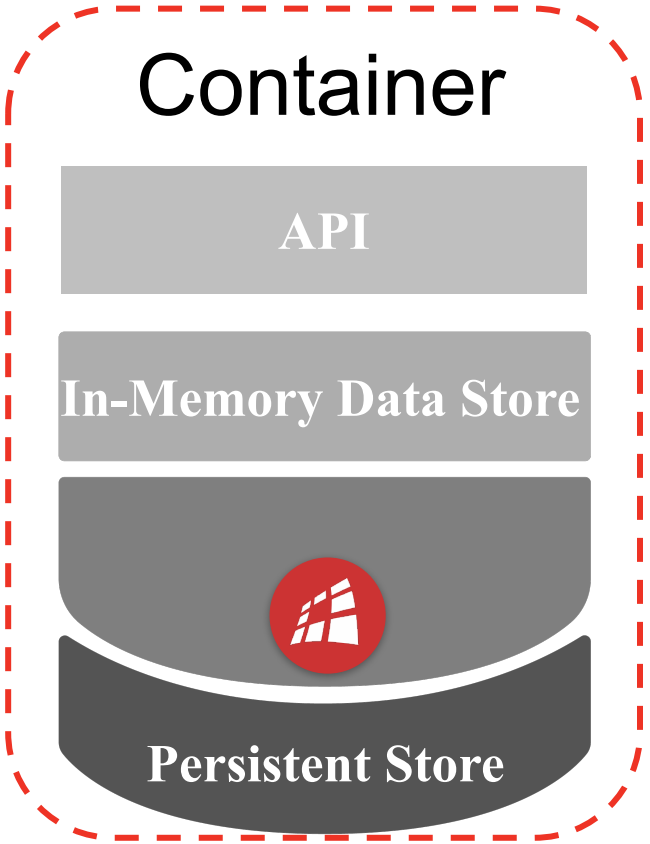
3 Shades of (in-process) HTAP



GridGain

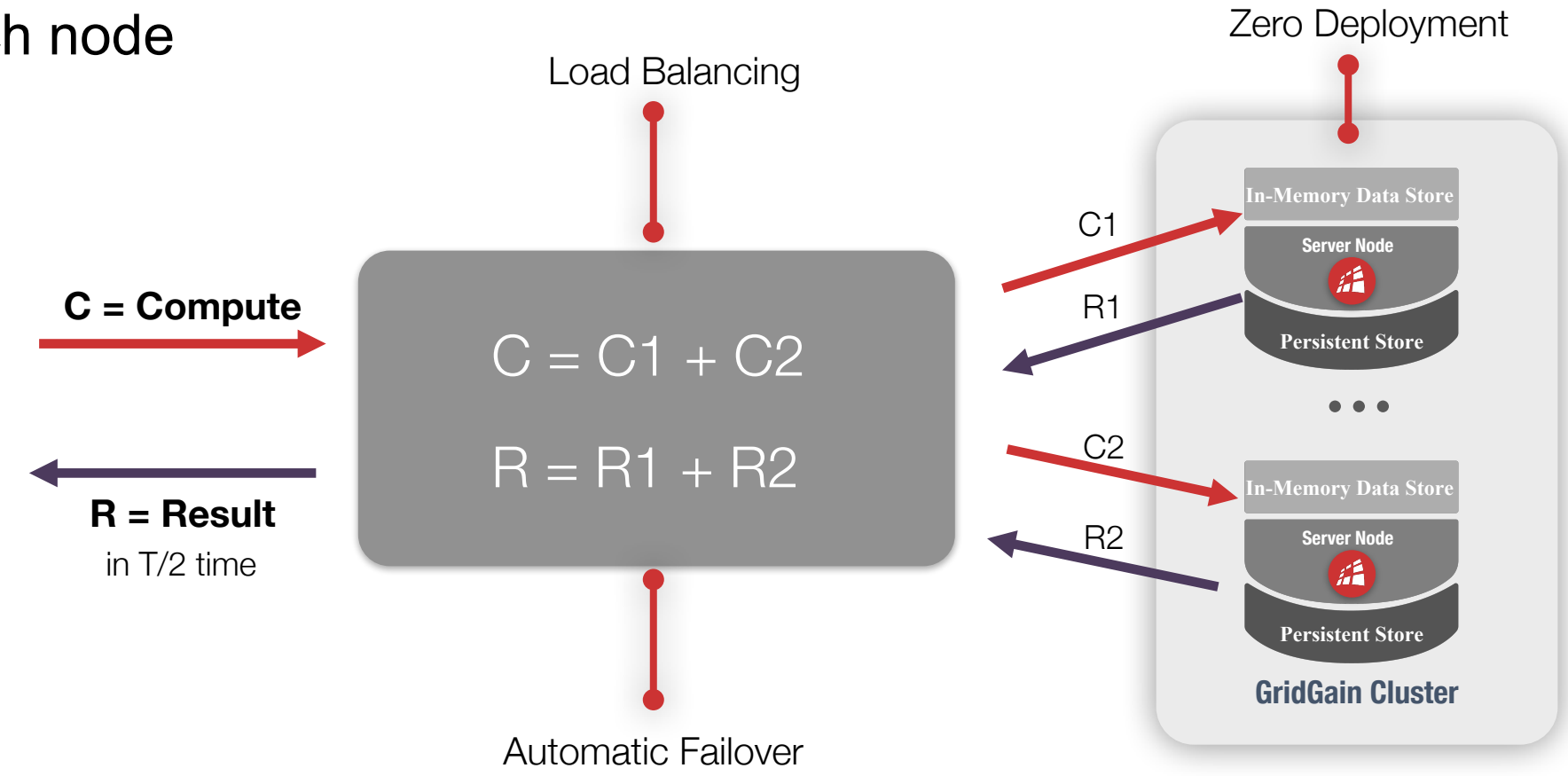
API-Centric HTAP – The First Phase

- Collocate data in-memory with each API
- Fit into the API deployment lifecycle (Docker / Kubernetes)
- Initialize data on startup with API initialization for elastic scale
- Simple data affinity/partitioning



Compute Grid – Data-centric, Real-time Analytics

- Distributes code to each node
  - Java, .NET or C++
  - Aggregates results
- Reduces network latency / bottleneck
- Separates analytics from API lifecycle

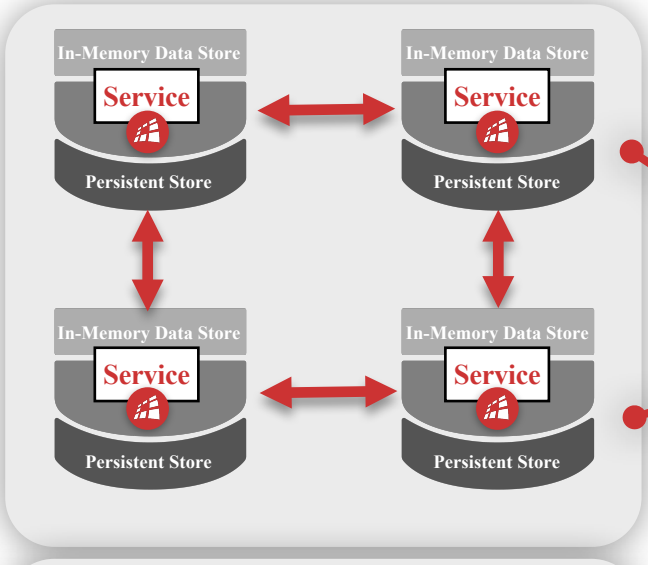


GridGain

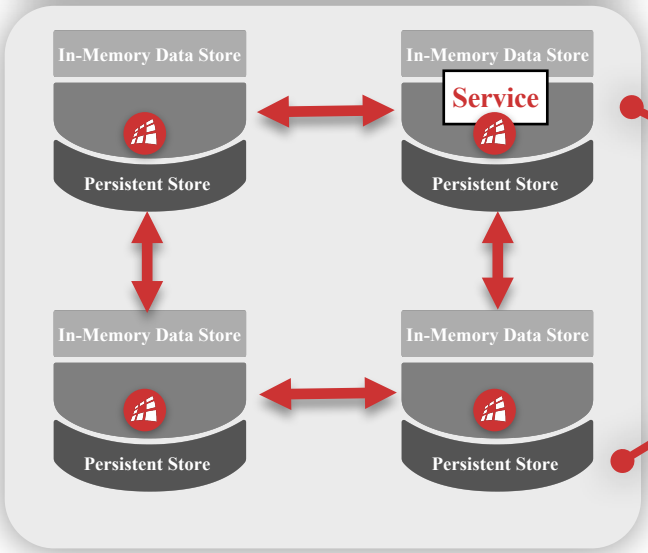
Service Grid – Data-Centric API Lifecycle Management

- Distributes RESTful API to each node
  - Java, .NET or C++
  - Aggregates results
- Leverages Compute Grid capabilities (collocation)
- Supports Singleton and scale-out patterns
- Allows APIs closely tied to data to be part of same lifecycle as the data

Cluster Singleton



Node Singleton



Best Practice 3.5 (2.6): Design with HTAP in Mind

- Plan ahead for HTAP
  - Real-time Analytics
  - In-process HTAP
- Identify which types of HTAP are needed
  - API-centric deployment
  - Data-centric APIs (Service Grid)
  - Data-centric analytics (Compute Grid)
- Plan when you need to build out each type

In-Memory Computing Best Practices Series

1. Build an in-memory computing foundation and roadmap
2. Add speed and scale to existing applications
3. Develop new applications, APIs and analytics
    1. Design around the customer experience
    2. Build a new real-time layer on top of existing systems
    3. Build real-time tasks, events, data and processes
    4. Free your data! Now!
    5. Design with HTAP in mind

In-Memory Computing Best Practices Series – Next Up

1. Build an in-memory computing foundation and roadmap
2. Add speed and scale to existing applications
3. Develop new applications, APIs and analytics
4. Make the bridge to the cloud
5. Implement HTAP and real-time data management
6. Become event- and stream-driven
7. Think different with machine and deep learning

GridGain

In-Memory Computing Best Practices

Developing New Applications, APIs and Analytics

Rob Meyer

Outbound Product Management

January 24, 2019

GridGain Resources

- Webinars
  - Visit https://www.gridgain.com/resources/webinars
  - Visit https://www.imcsummit.org/
- White Papers
  - Visit https://www.gridgain.com/resources/papers
- Videos
  - Visit https://www.gridgain.com/resources/videos
- Free 30-Day Ultimate, Enterprise or Professional Edition Trial
  - Visit https://www.gridgain.com/resources/download

GridGain