RITdb

Data Infrastructure for Backend Smart Manufacturing

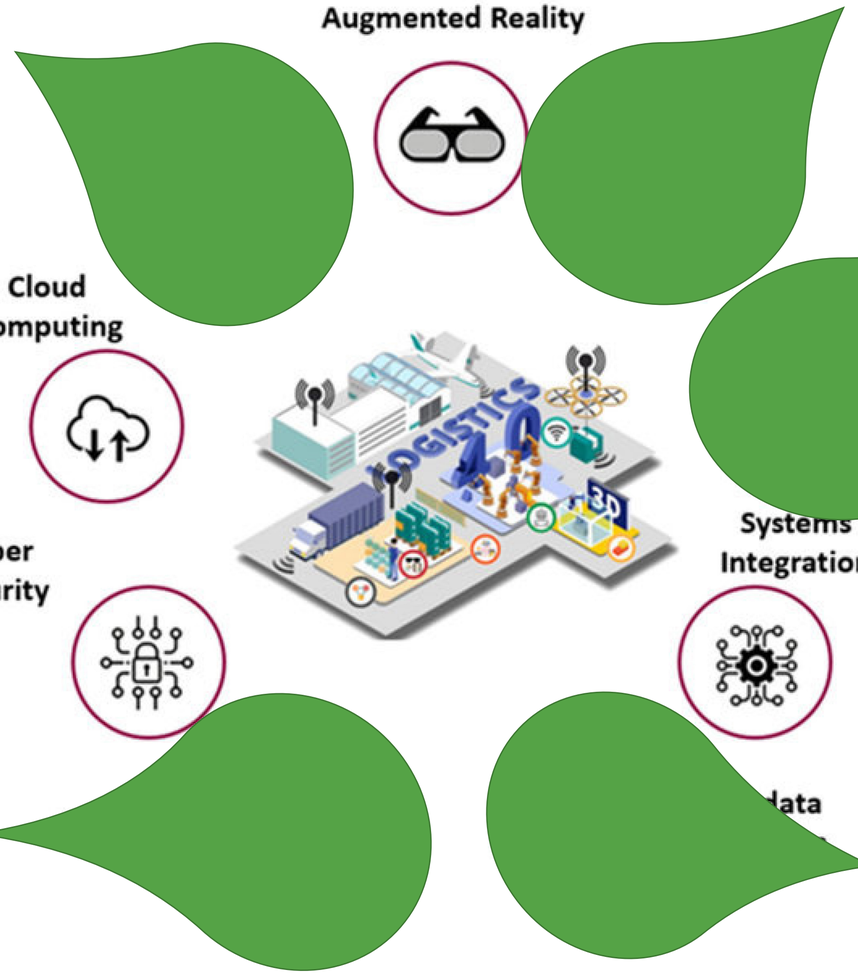
RITdb Task Force

Stacy Ajouri, SMTS — **Texas Instruments**

Mark Roos, CEO — **Roos Instruments**

Main Pillars of Industry 4.0

HIR
Traceability



MQTT/CBOR

Smart Test Cell ++

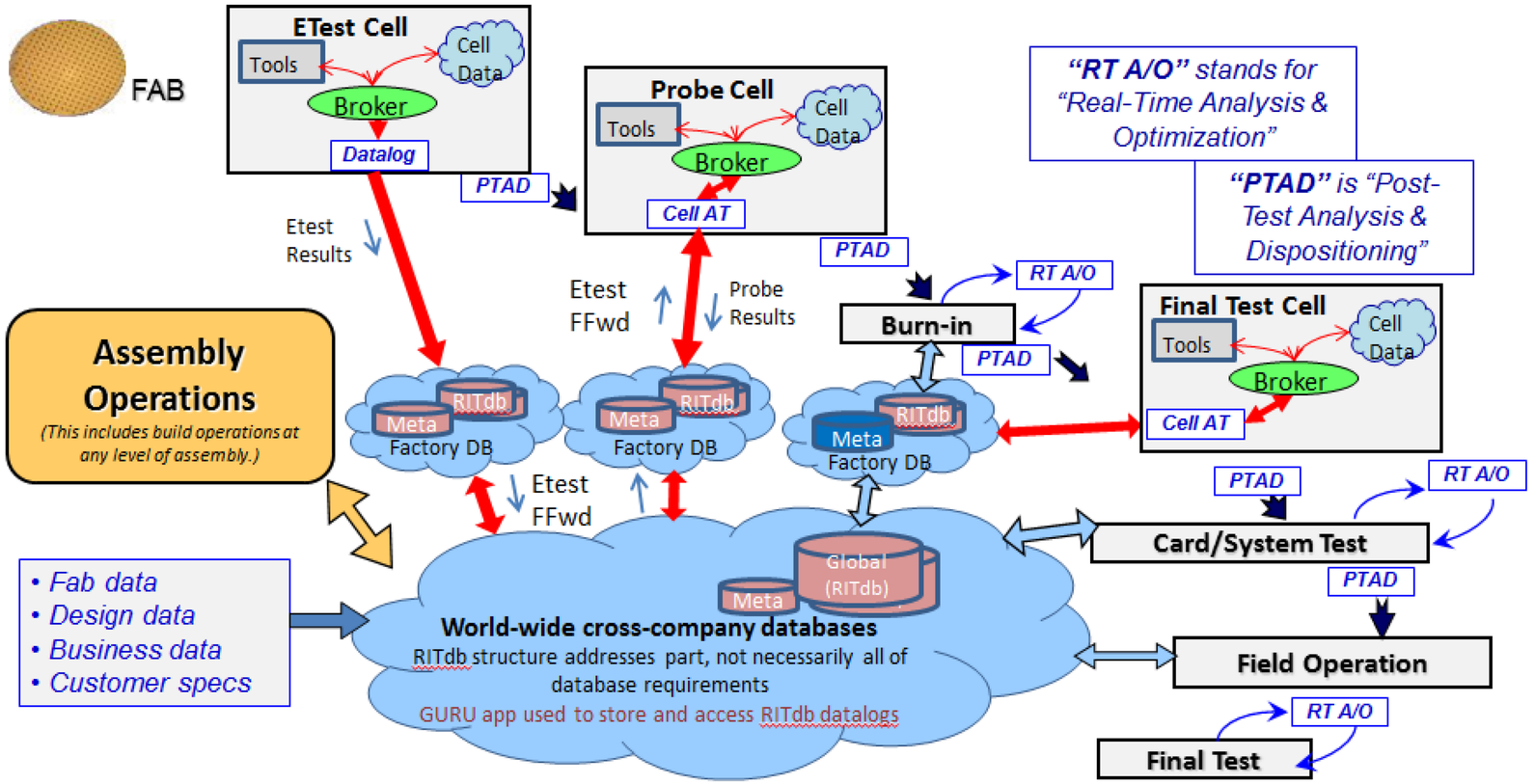
Replay
Engine

ETL - provisioning data that is ready to use!

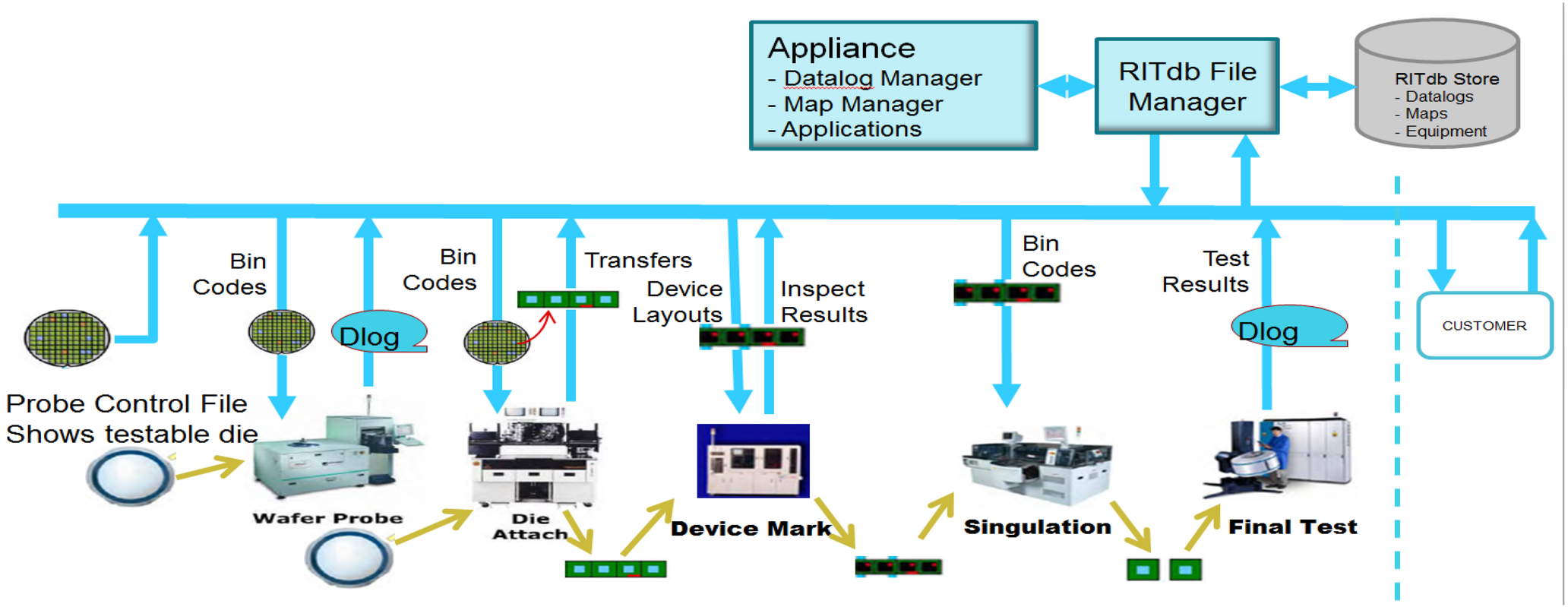
Expectation of Smart Mfg for Backend

- Faster response time
- Improved Quality
- Improved Utilization
- Democratization of the data
  - Test Floor is a data lake…
  - Easy insertion/easy use
  - More data sources for more precision decision
  - Better Monitoring
- Traceability
- Integration of Legacy Equipment

Big Data Universe



Backend Data flow



timeline

How did we arrive at RITdb

2007-2012

- 2007 SEMI STDF Standard published
- 2010 Aligned with ITRS Adaptive Test Working Group
- 2011 **CAST Survey** Proposal to use SQLite
- 2012 – CAST WG: Next Generation STDF.

2014-2016

- 2014 Start of document and creation of first translators
- 2015 Start of RITdb and TEMS Alignment + Inclusion of E142 Model
- 2016 RI POC for STDF to RITdb
- 2016 **TI POC for OEE Streams** Start of Streaming RITdb

2017-2019

- 2017 TI POC ATE Equipment Resource Tracking
- 2018 **Xcerra POC** for container and **validator**
- 2019 RITdb Task Force
- 2019 Advantest POC for native RITdb with TI

2020-2021

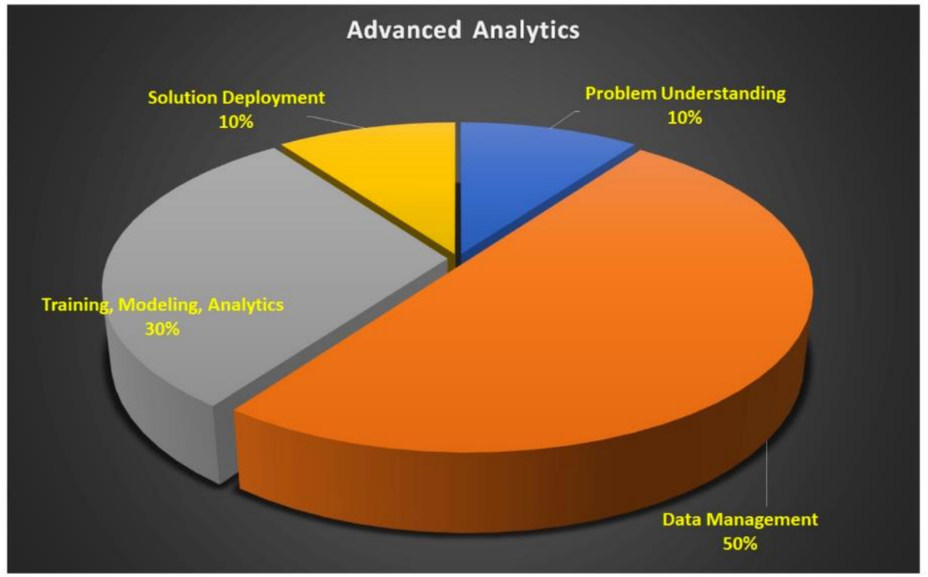
- 2020 PCM Test Study Expert System for Rule Evaluation
- 2021 **Genealogy Tracking** Experiment
- 2021 First SEMI Standard for RITdb

Cycle 4 Ballot

Biggest Concerns when implementing a Test Cell

CAST Member Survey (2011)

1. Data
   - ETL necessary for the wide variety of data in the back end. 60+%
   - First focus is on data availability.
   - Shared data model will reduce the ETL issue
2. Adaptive Test
   - Issue is the scope of the cells and variation of flow.
   - Wide variability of algos, accessing the data, & response time.



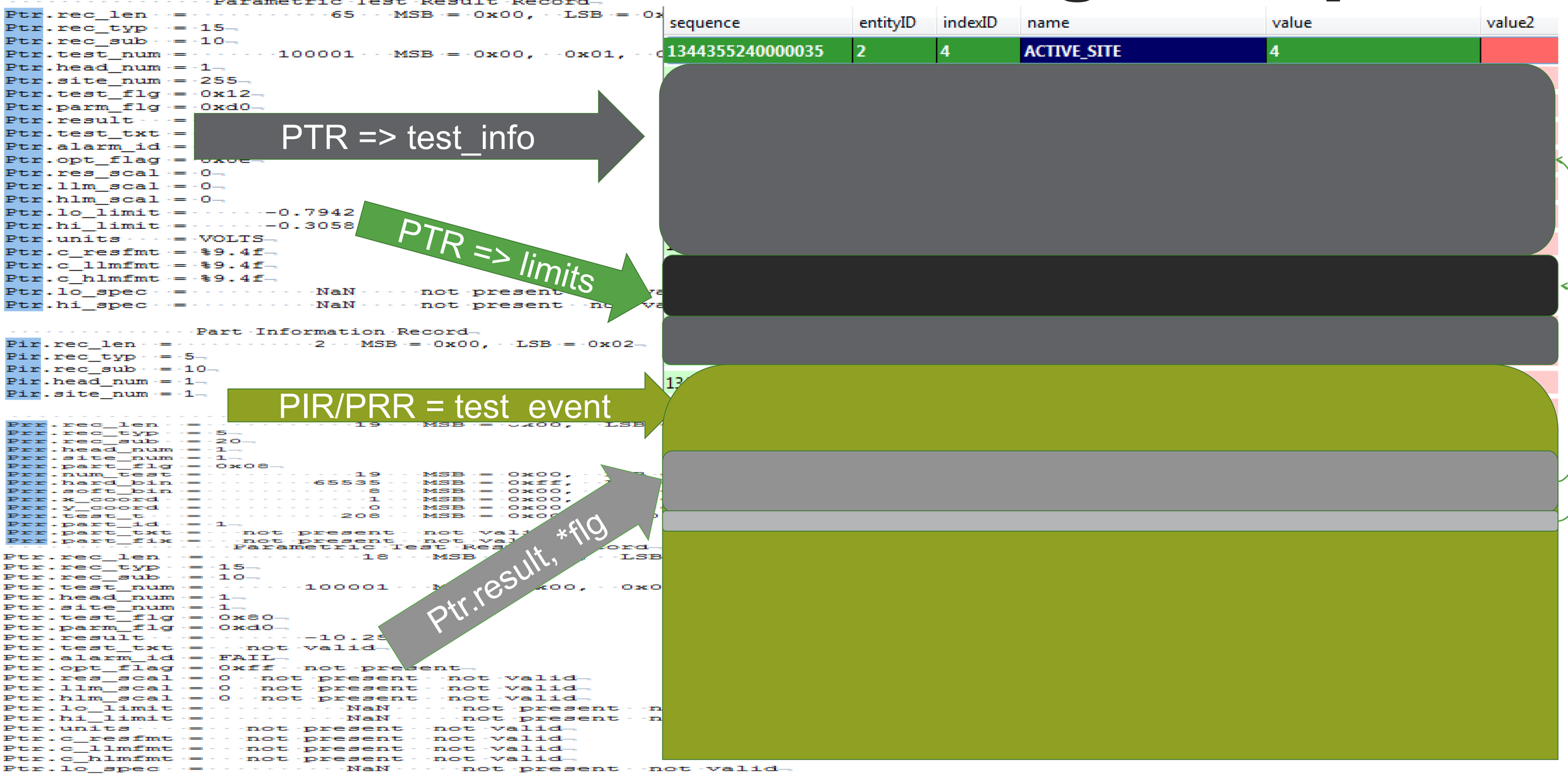
Opportunity identified was for data management.

#1 Priority

RITdb Container

- SQLite
- Can be queried using SQL
- No SQL narrow table
- Can be machine validated against spec.
- Attached metadata for provenance
- Validated with many STDF examples

RITdb Data Container: datalog example



Provenance

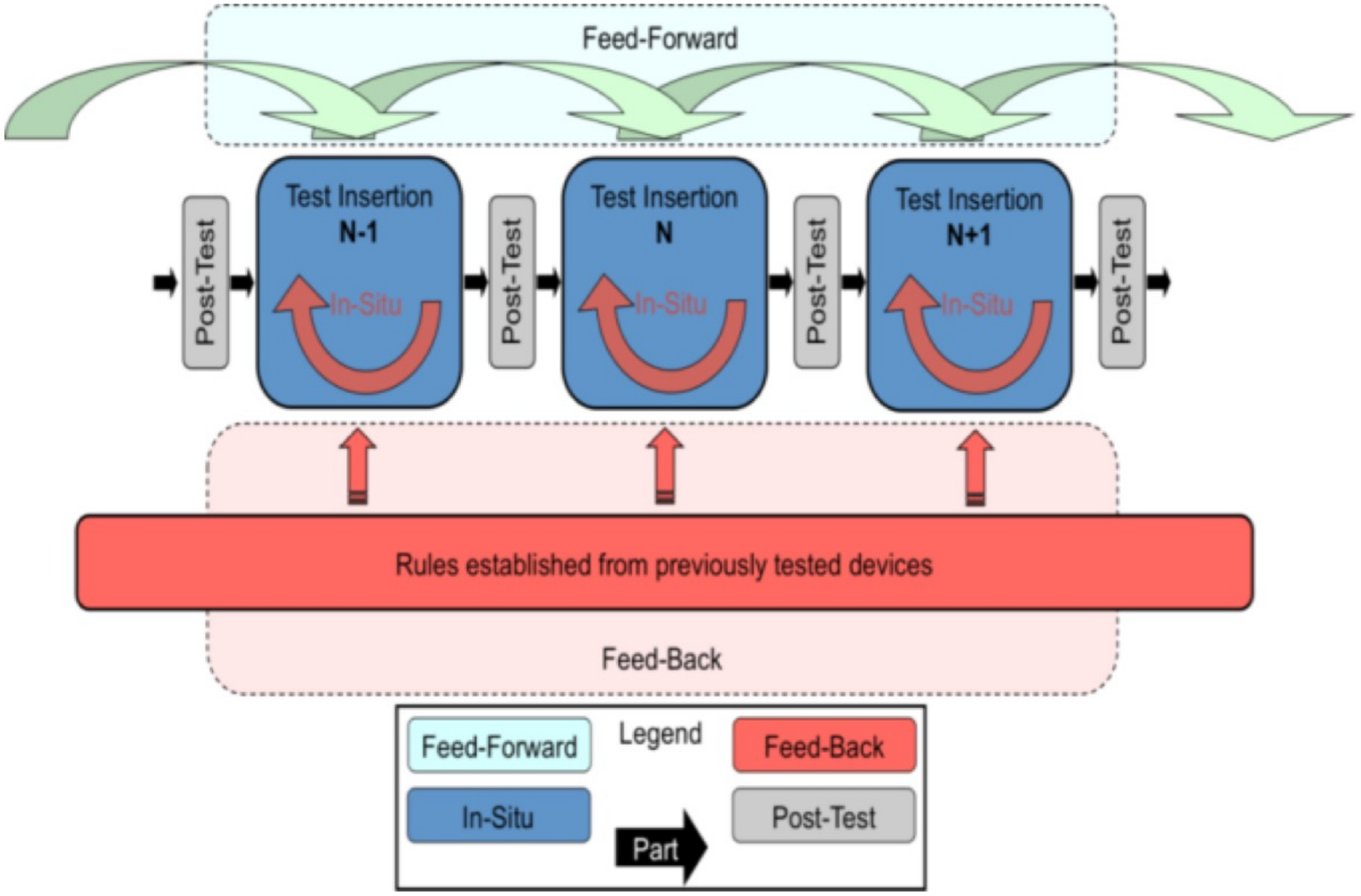
- Source Identity
  - Who created the data
- Integrity
  - Is it the data they created
- Security
  - Who can access the data
- Classification
  - Grouping by some characteristic
  - Source, content, type, product,time,….
- Lineage
  - Where it has been and what happened to it

Evaluation of Other Data Types

- Validated in POC
  - Probe
  - Final Test (Singulated/Strip)
  - Equipmentlog (ATE – 10+)
  - Traceability
  - OEE Events
- Designed – to be tested
  - E142 (HIR)
  - PCM (WAT, inline electrical)
  - SLT
  - Equipmentlog (other)
  - Large digital Test Data
  - AI/ML structures

#2 priority

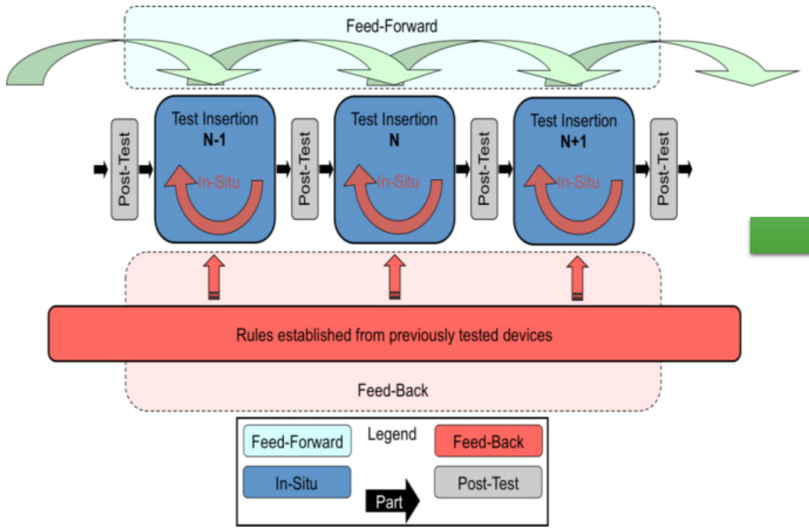
Adaptive Test



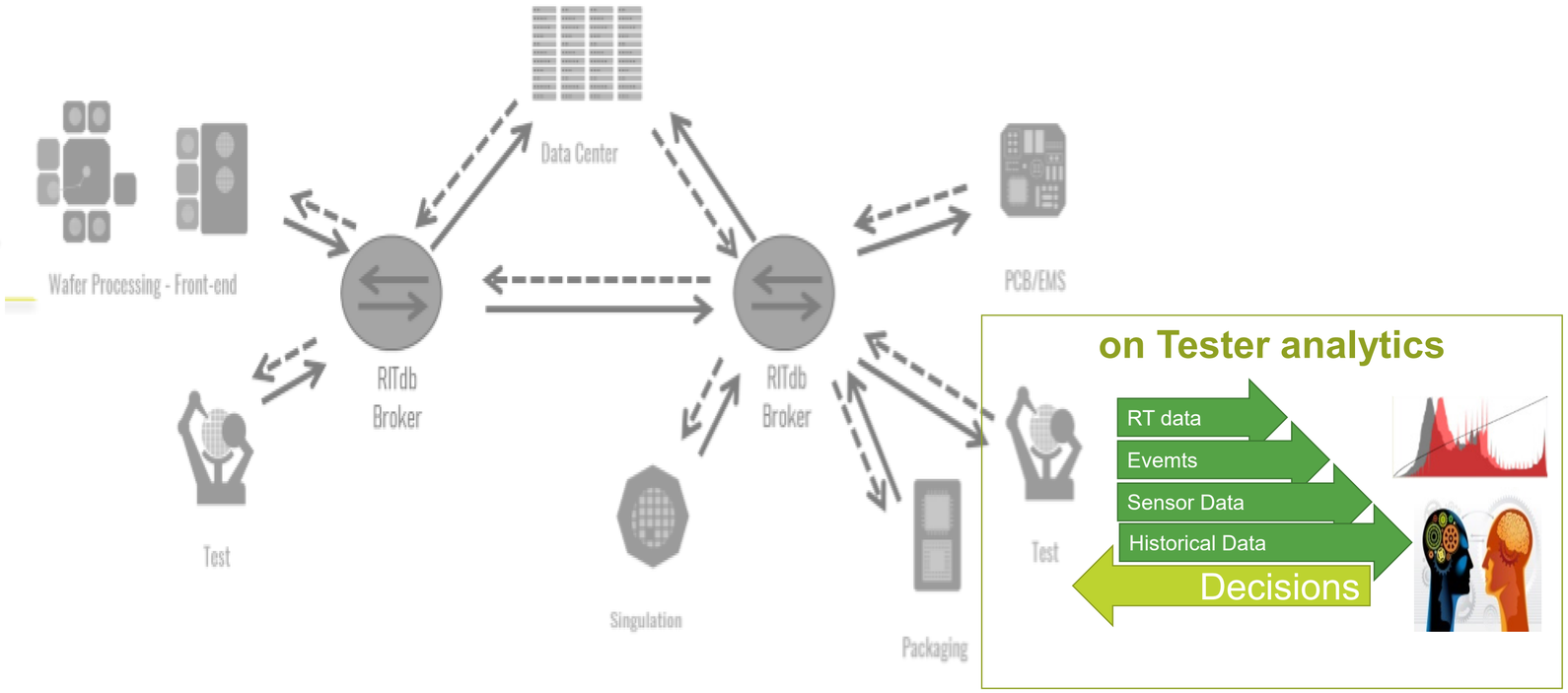
Streaming RITdb is a flexible, extendable system with the ability to collect and act upon data in real-time.

- Real Time data system modeled on M2M IOT enabling Plug and Play for tools and test support applications. Data is bi-directional.
- Standard addresses data streaming, transport and storage. i.e. supports generation and access to real-time streaming data as well as batch data (history)
- Allows for data from different producers to be merged and synchronized, and then delivered to consumer. Can be run in parallel with existing systems.
- Supports data security and integrity as well as allowing addition of new types of data without impacting existing model
- Data is distributed, integrates with Adaptive Test Model and Big Data

Industrial Internet of Things enables Adaptive Test



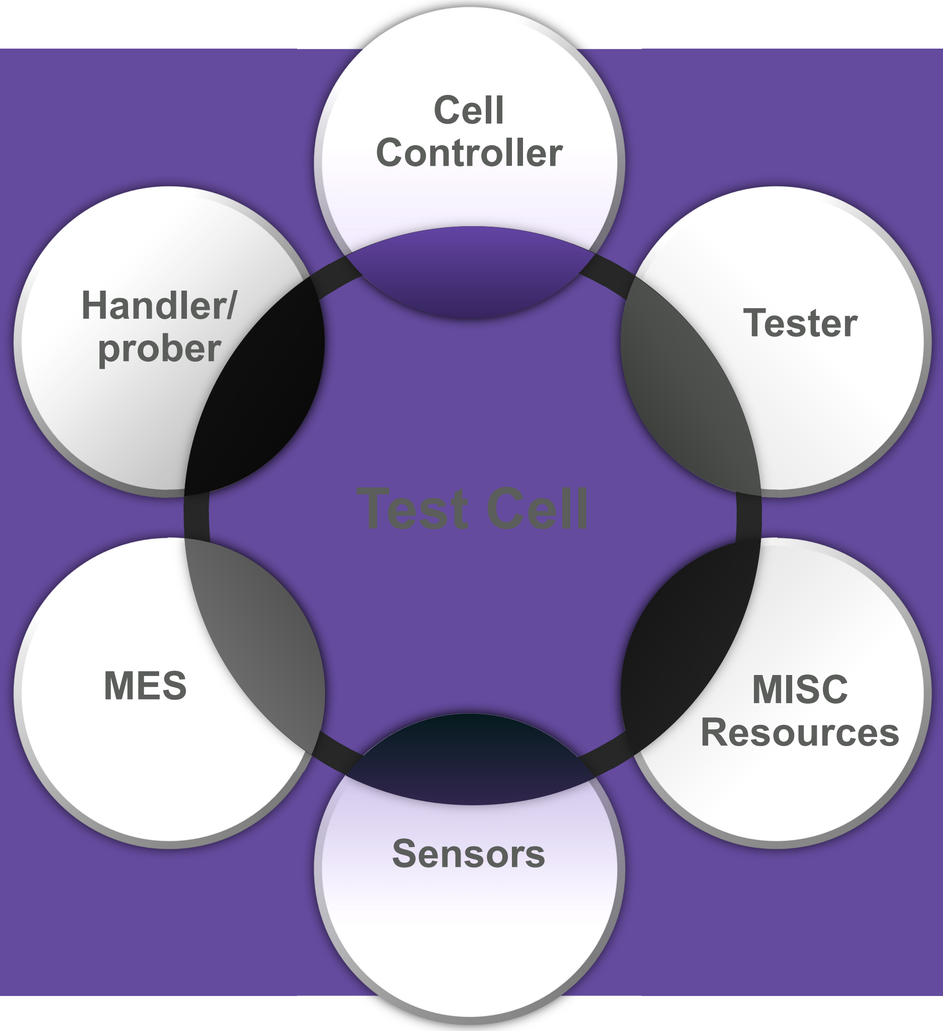
IEEE Electronics Packaging Society HIR (Heterogeneous Integration Roadmap)



Events come from many sources:

ATE, OEE, Handler/Prober, Test Cell Control
MES, Automation Systems, …..

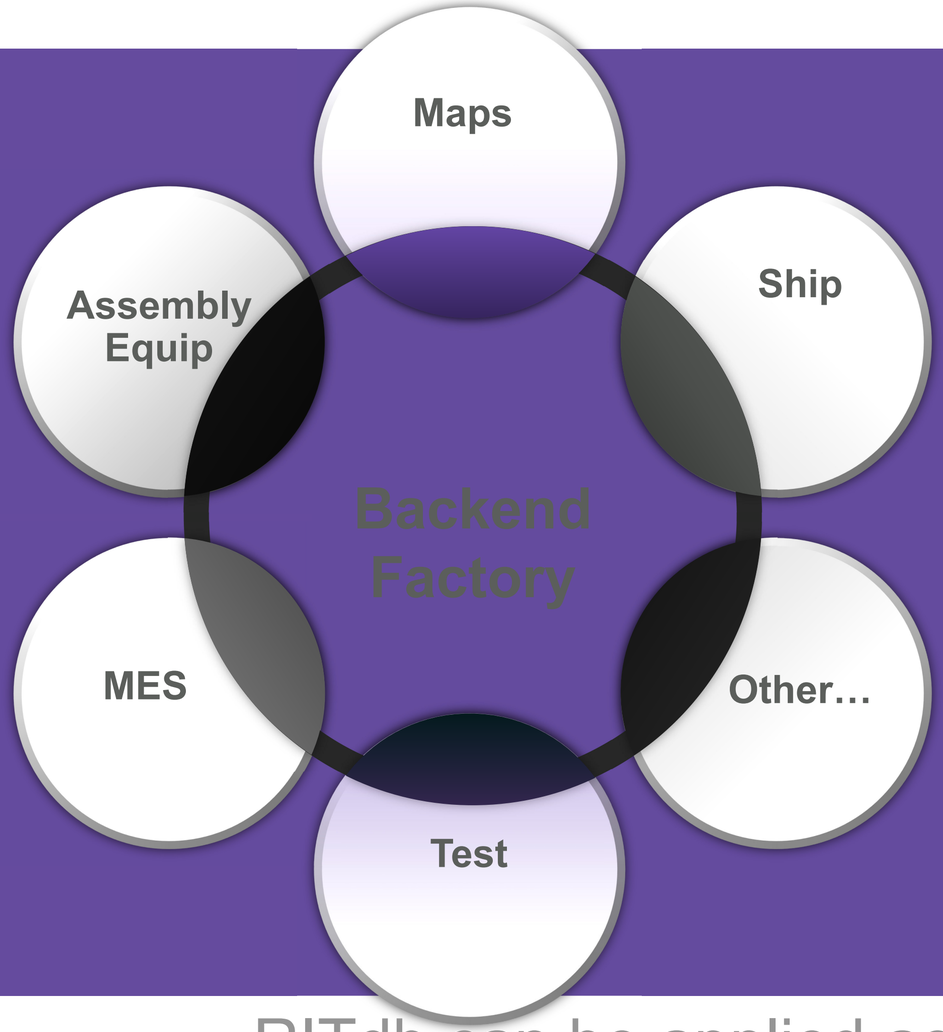
RITdb Opportunities:



Add ability to stream information from all parts of a test cell.

- Cell controller – record key information/decisions from cell control.
- Handler/Prober – monitor critical settings, recipe setup,…
- Tester – Monitor tester diagnostics/calibration and integrate with operations when potential issue is identified.
- Sensors – Integrate sensor events with test events.
- *And much more….*

RITdb Beyond the Test Cell:



Add ability to stream information from anywhere

- Record Maps and product transformations. i.e. wafer die to Strip location, Strip to Tape & Reel, die to multi-product packages,....
- Generate events across backend flow. Allow better alignment between probe, assembly and test.

*** Backend: from PCM to Ship

RITdb can be applied across the eco system not just the test cell.

*A/T = Assembly & Test

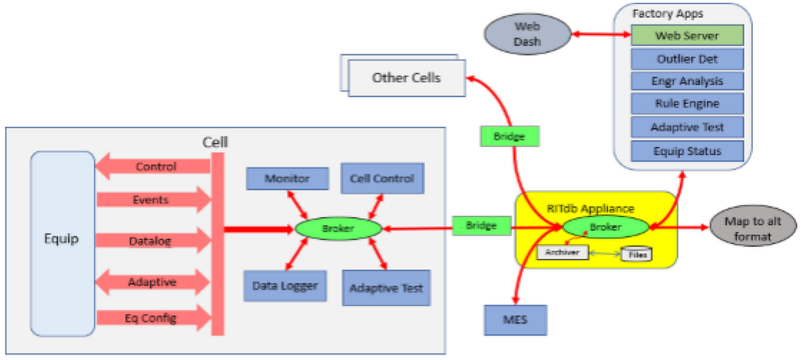
RITdb – A foundation for Smart Manufacturing Backend Data

- RITdb Features
  - RITdb Containers and/or streams of data
  - Capable of integrating batch data with individual streams
- RITdb bi-directional messaging
  - Exchange information between Test cell and anything else connected to RITdb (data or control)
  - Enables operational control and data analytics

Build upon IoT architecture using MQTT messaging protocol

- Modern open source message based communication enables plug and play tools and applications
- Adds layer of security with private/public key sharing rules
- Easily integrates into Big Data infrastructure

Logical Model for a Test Cell RITdb Environment



Red lines - MQTT channels which carry RITdb messages
Dark blue - examples of cell and factory functionality
Brown - legacy tool support

POC - Applications

POC (completed/active)

- RITdb Tools
- ATE Equipment Resource Tracking
- Data Visualization
- Replay engine
- Rules Engine
- Traceability
- ATE Clients (JAVA, C++, Python)

POC (Planned)

- SLT
- Bench Testing
- Cell Topology
- Data Analysis Scaling
- Failure Recovery/Updates
- Security/Encryption
- Big Digital
- AI/ML
- Robot Integration
- Cloud Integration

Next Steps for RITdb Standard

- **RITdb Ballot - Cycle 4**
  - Voting Period Starts: Wednesday, April 28, 2021
  - Voting Period Ends: Friday, May 28, 2021
- Event Message Standardization
- Machine Learning Support

Where should we go next?