Legacy Systems and Programming for Devices
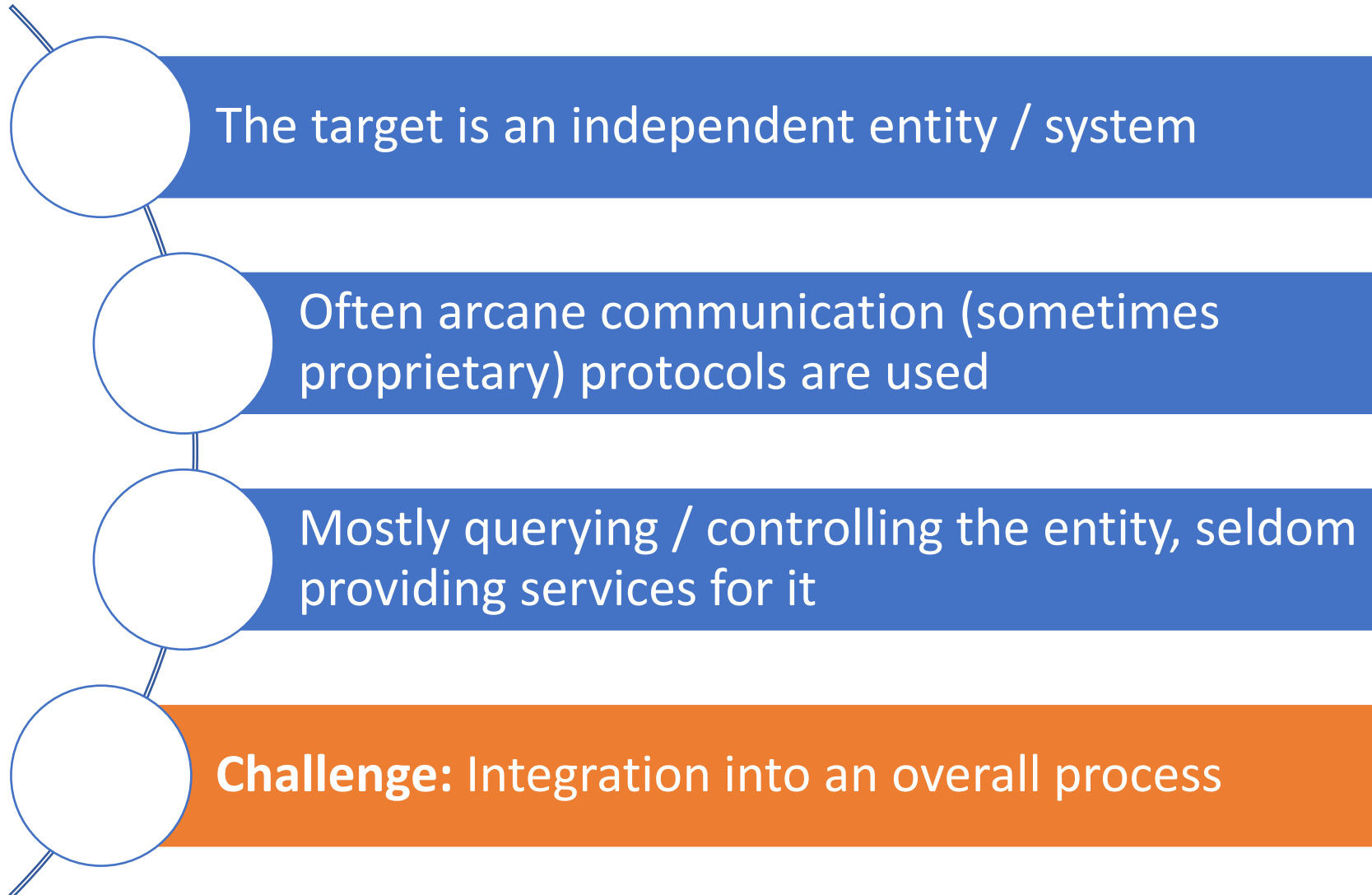
Michael Glienecke
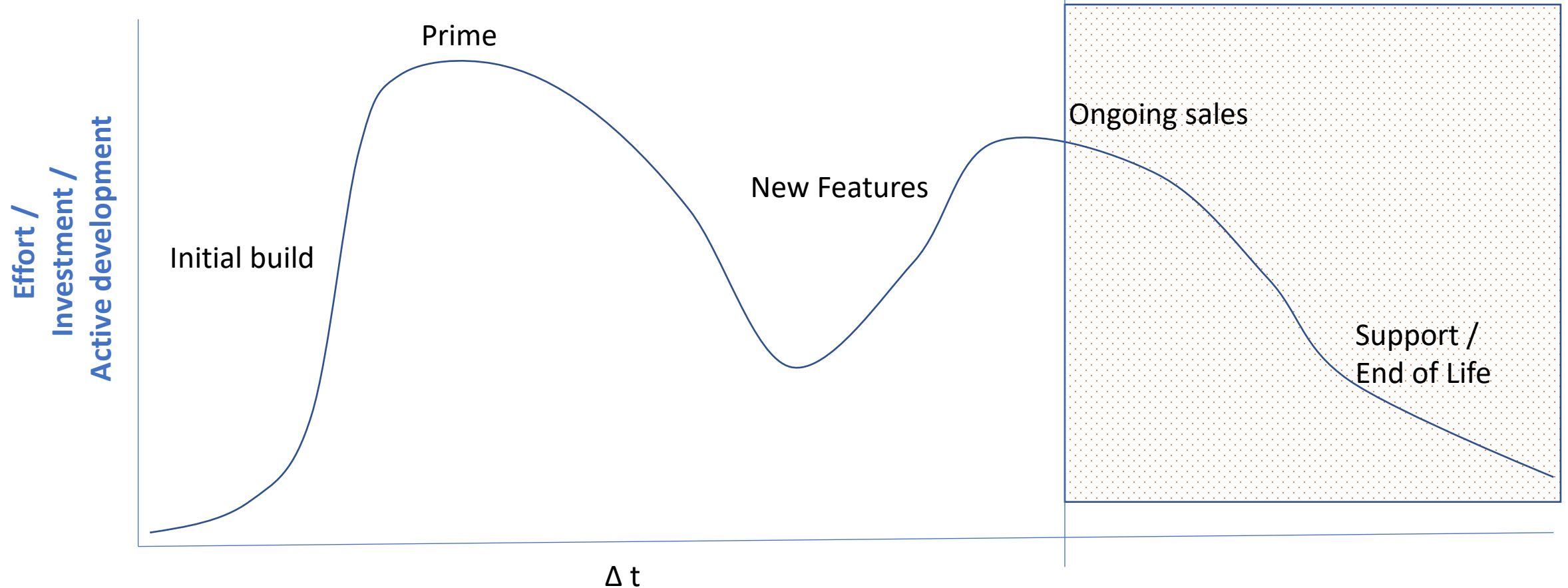
Legacy Systems and Programming for Devices have many things in common

The target is an independent entity / system

Often arcane communication (sometimes proprietary) protocols are used

Mostly querying / controlling the entity, seldom providing services for it

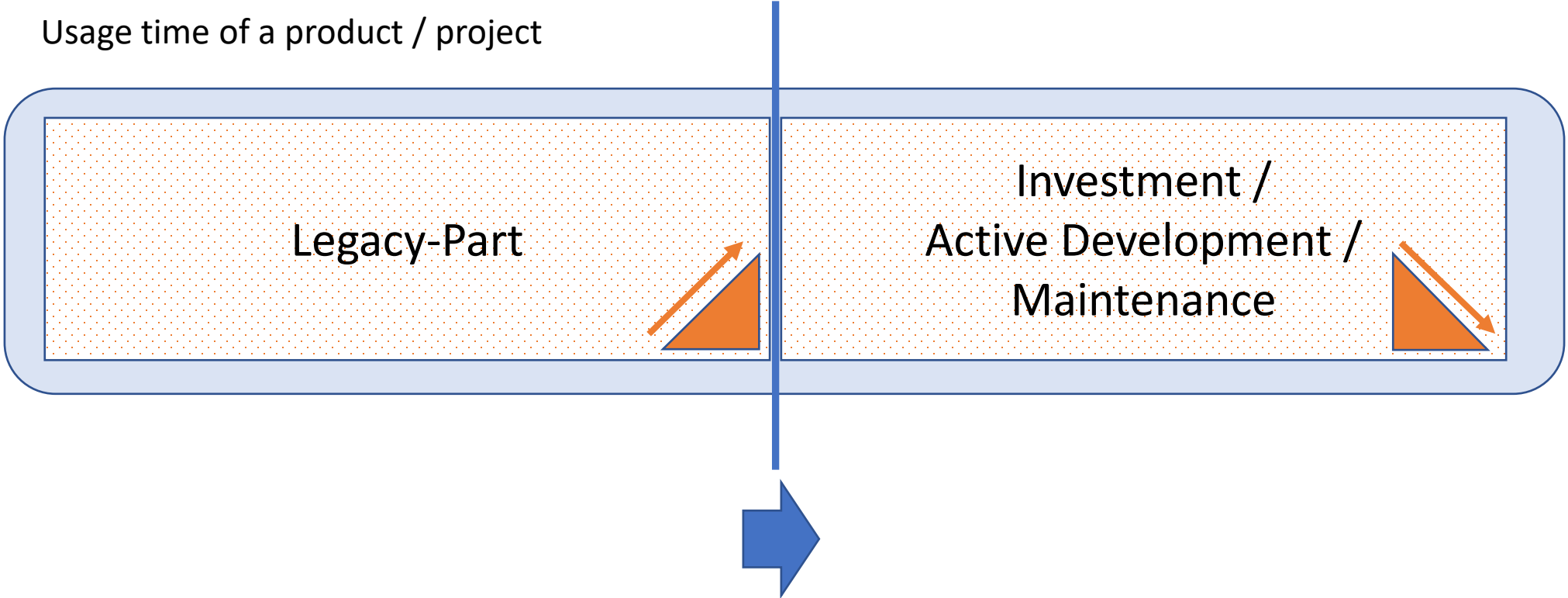**Challenge:** Integration into an overall process

So is legacy only about mainframes and some weird old systems?
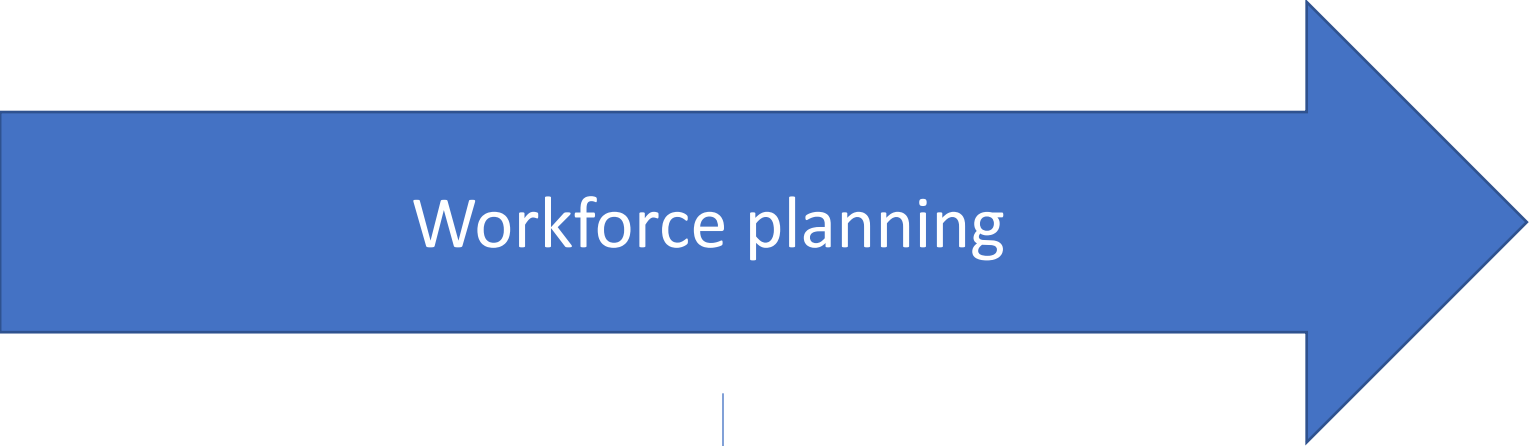# No - Legacy is anything which is no longer under active development -> technical dept

Typical lifetime of a product

Legacy starts

Effort / Investment / Active development

Prime

New Features

Initial build

Ongoing sales

Support / End of Life

Δ t

Any (used) project / product will become legacy (a technical dept) over time

Usage time of a product / project

Legacy-Part

Investment /
Active Development /
Maintenance

The big question is how to integrate legacy systems.
A real-world Example

The most common legacy integration technologies

CSV / File-Transfer

MQ-Series

SSH

"old" Web-Services

SAP R/3

CICS

(Micro)-Services (REST, SOAP, gRPC)
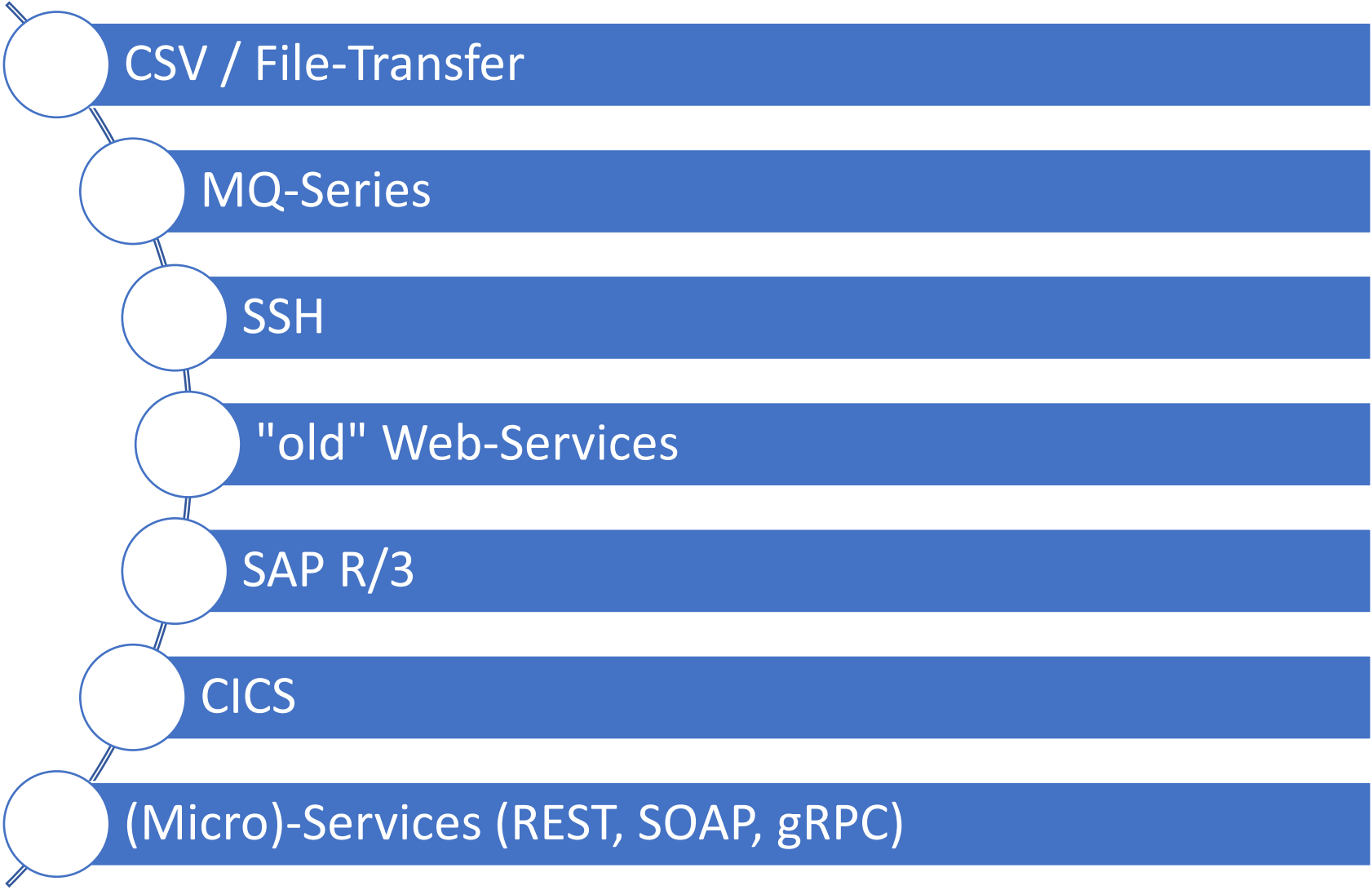
Device integration has many facets to consider
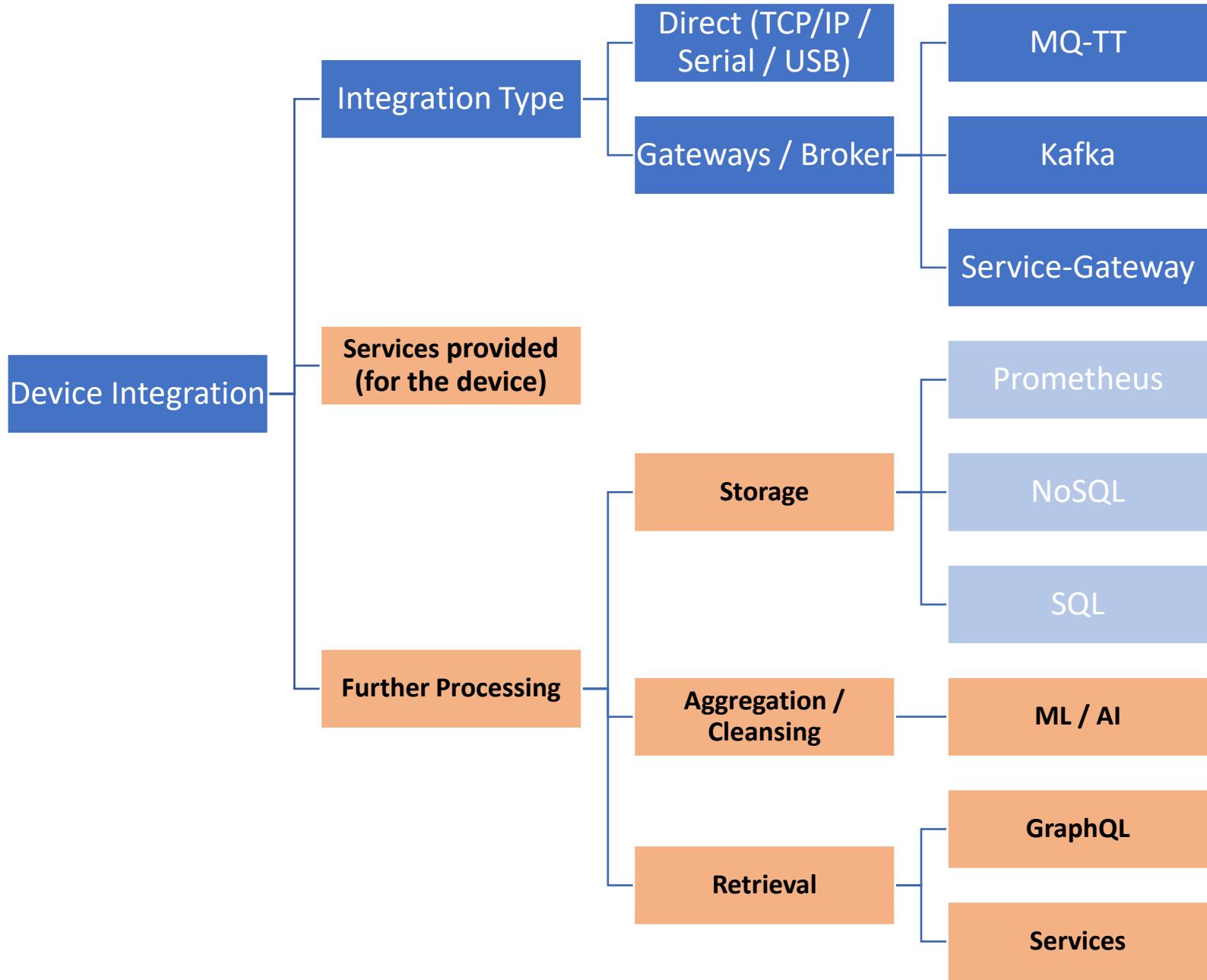
Example of integrating devices (corrosion center / production feedback workflow) and legacy (production process)
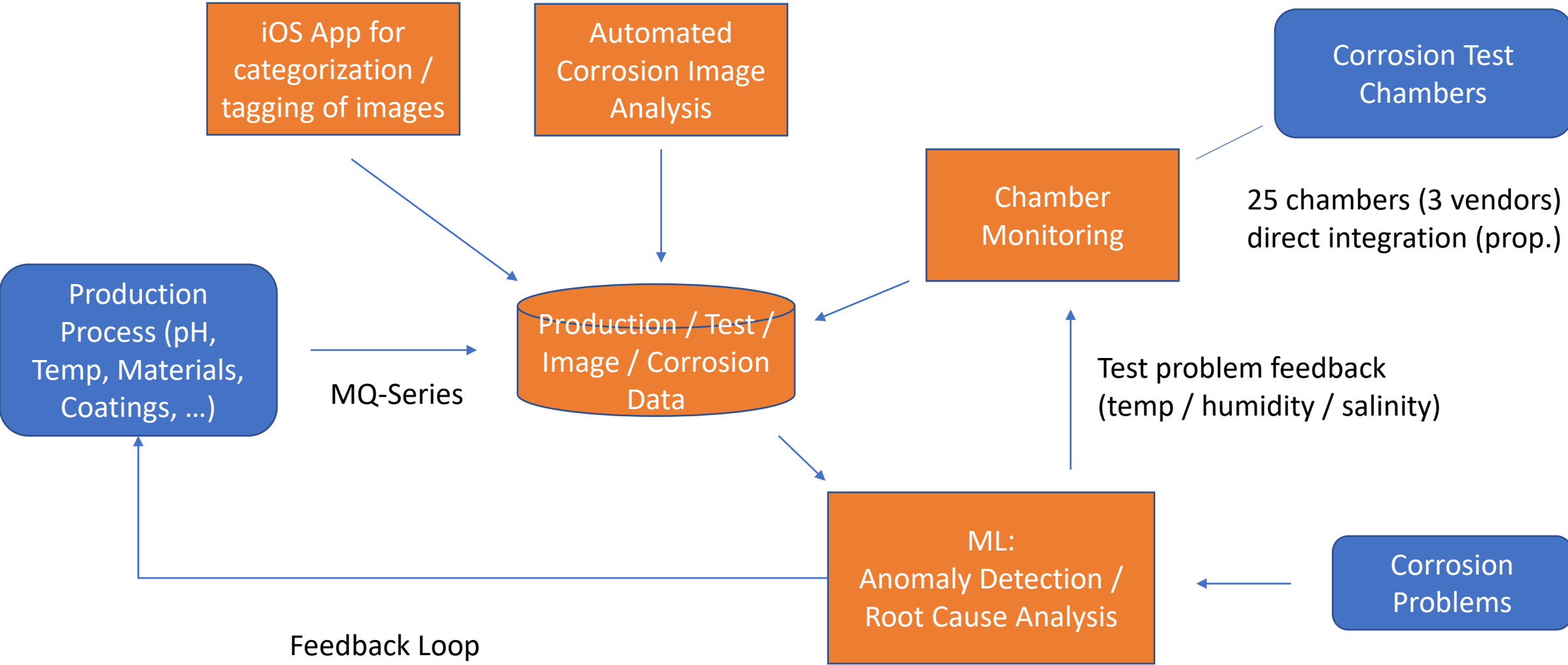
The Take-Away... Focus less on technology, more on use-case, integration and process

- Protocols, hardware, technology can be fixed and solved
  Nonetheless, good craftsmanship is important

- Business processes (-> Industry 4.0) are paramount

- Use (Micro)-Services for integration