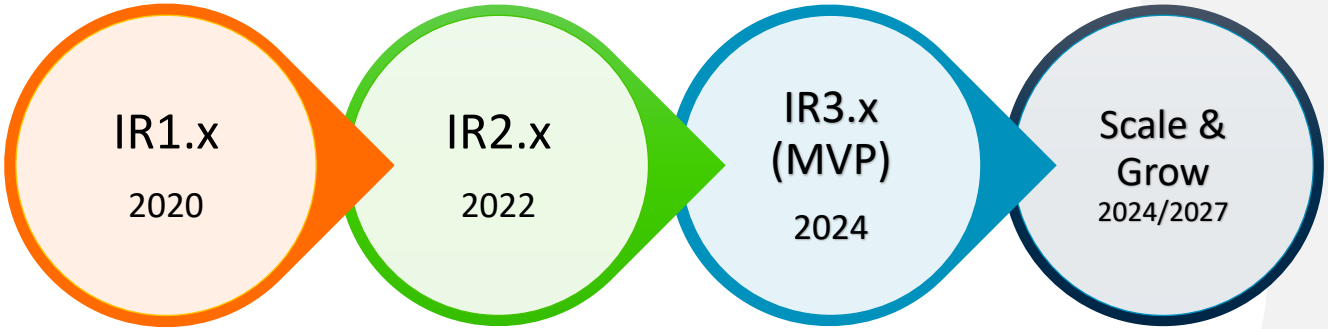
SystemReady Devicetree band

FY25 and FY26 Roadmap

arm SystemReady

Within the aspiring “SW Just Works” vision, SystemReady’s mission is to provide off-the-shelf inter-operability between OSs and Platforms

SystemReady IR Iterations - Goals



(IR1.1) Kick start: laid the foundations of EBBR upon existing UEFI, DT standards. Certification Scheme and Compliance Suite was created for **basic** capabilities testing

(IR2.x) Correctness of those basic capabilities as:

- FW update (ESRT, GUID)
- DT conformance
- Capsule authentication
- Integration of SIE into ACS

(IR3.x) completeness of requirements that effectively delivers the SystemReady promise through:

- Boot capabilities
- Maintenance capabilities
- Security capabilities

Transition to an adoption model that scales by fostering SR-DT integration into vendors flows and self-compliance

Grow specs up to accommodate new industry use-cases and demand

A model that scales

Pivot to compliance

- **Motivation**: scalability and sustainability
- **Challenge**: The certification process and ACS were originally designed for Arm's certification engineers to execute and validate, with test houses assisting providing logs but Arm inspecting and issuing certifications. Enabling vendor to self-declaration of compliance required chances to both process and tooling.
- Staggered approach
  - ✓ Phase 1: Autonomous testing. (*released in November'24*)
    - Compliance determination does not require human interpretation
    - Only qualified test-houses can handle the compliance process (until tooling is suitable for general use)
  - ✓ Phase 2: Semi-automated testing. (*released in April'25*)
    - Test flow, suites, post-processing scripts and test integration automation
    - Manual intervention is reduced through configuration files and waiver files
    - Qualified vendors along test-houses are allowed to self-declare compliance
  - Phase 3: Suitable for general use (*no committed date*)
    - Further automation, tool stability and documentation quality
    - Any vendor can declare self-compliance

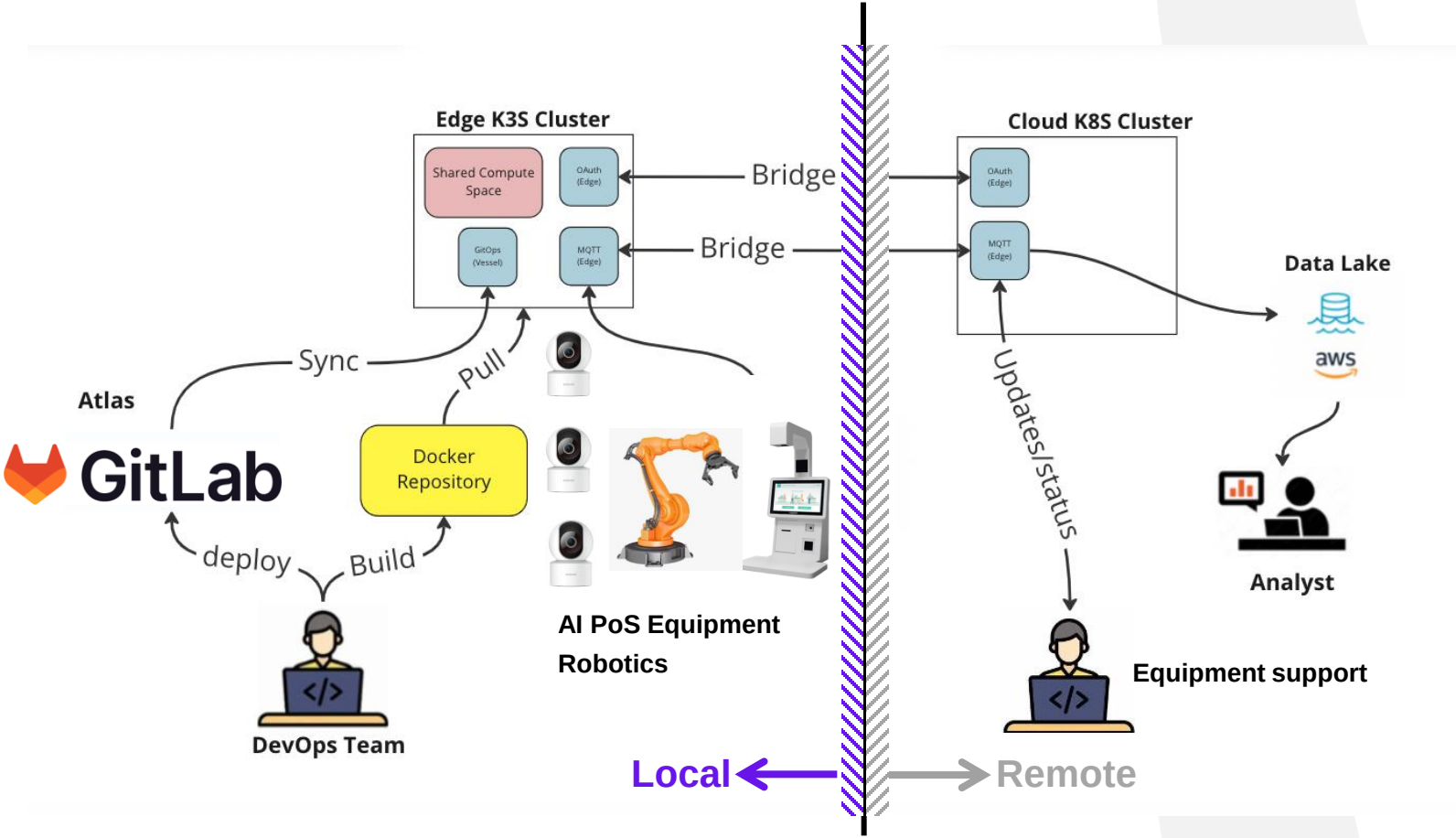
Grow -- capabilities & tests

Fundamental Capabilities

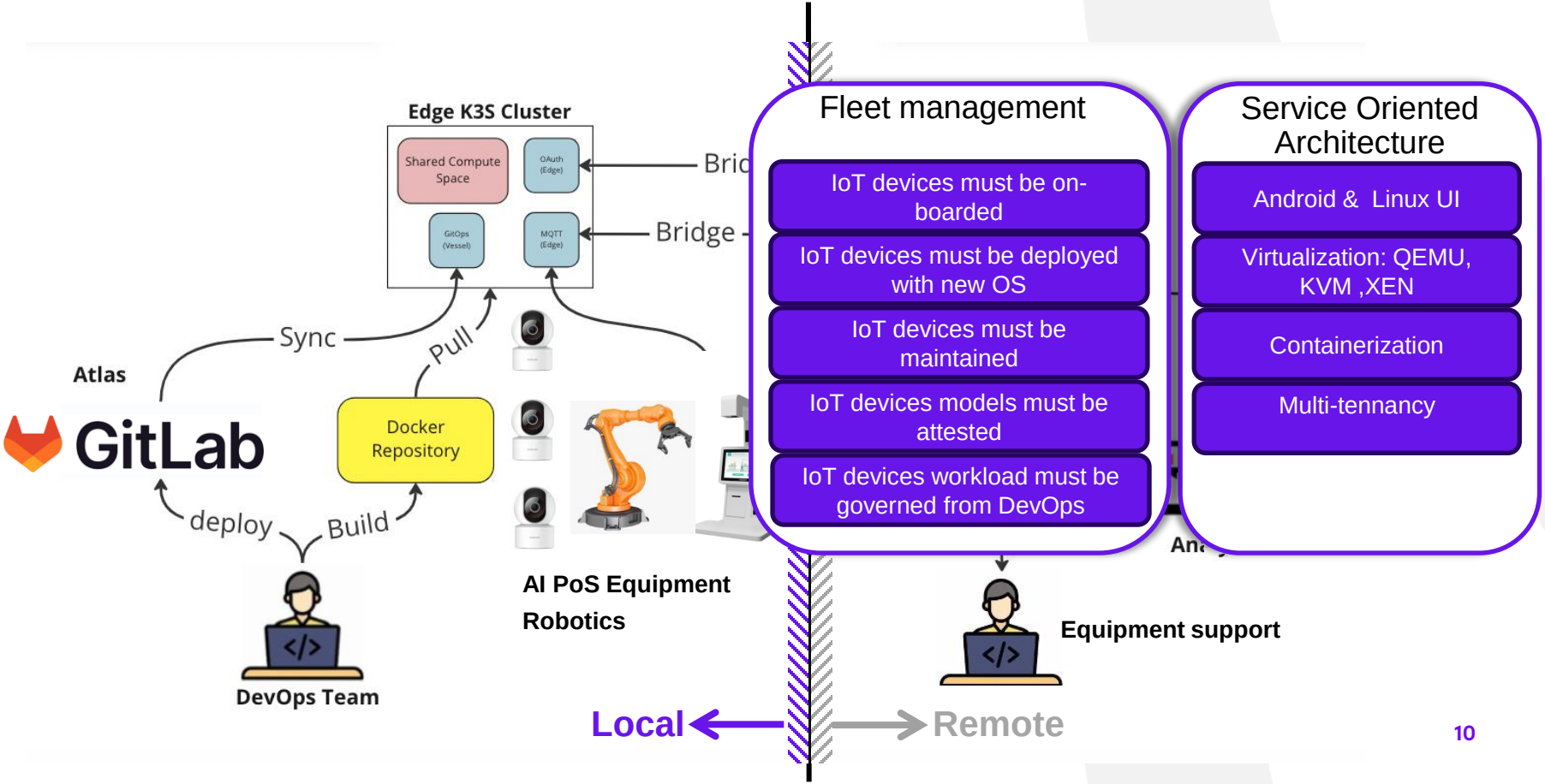
- Finalize SystemReady's DT MVP
  - ✓ Boot Readiness
  - ✓ Security Readiness
  - ⚠ Maintenance Readiness
    - A/B Support for capsule update

Edge IoT capabilities

On-prem Computing architecture (example retail / industrial ...)

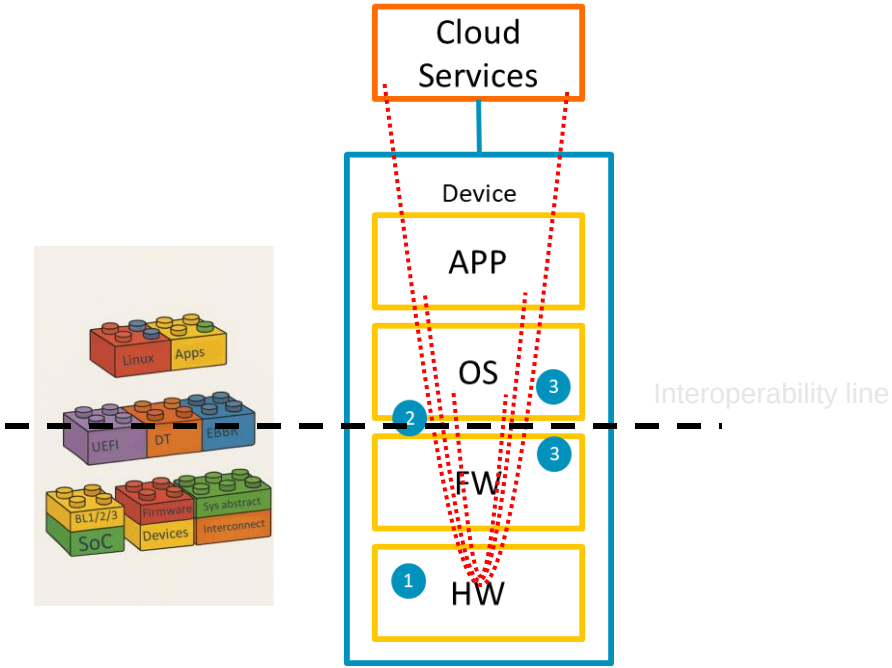


On-prem Computing architecture (retail / industrial …)



Reminder: SystemReady scope is limited to ….

SystemReady as a hygiene factor providing interoperable building blocks to others



- Interoperability line crossings are within the scope of SystemReady
- Services/Functions not crossing the interoperability line are not within the SystemReady scope
- Examples of in/out of scope:
  - ✓ AWS Greengrass provisioning keys to be stored in secure storage.
  - ✗ AWS Greengrass MQTT bridge to cloud services.
- SystemReady can be seen as a hygiene factor to upper layers in the stack providing a standard way to reach the platform

Gap analysis conclusion and recommendation

Opportunities and capabilities mature enough to tackle by SystemReady Devicetree

- Android support
  - Google working on Android's bootloader through UEFI and additional extensions
  - This is an opportunity for aligning both Android & SR-DT so a code base can land both Linux and Android.
- Virtualization
  - Although supported in the SR and ES bands from day one, there is growing demand for Devicetree devices to offer equivalent capabilities
  - Adaptation of good practices within the infra space into the more constraint IoT industry
- Secure session establishment (key provisioning) for device on-boarding
  - EU-CRA will regulate security for IoT devices operation, including on-boarding.
  - There's an opportunity to provide the building blocks to prevent fragmentation on how to provision keys to IoT devices and meeting EU-CRA

Next steps: Bring this topics to the wider SystemArchAC and potentially others for consensus.

*Gap analysis workshop recordings and no

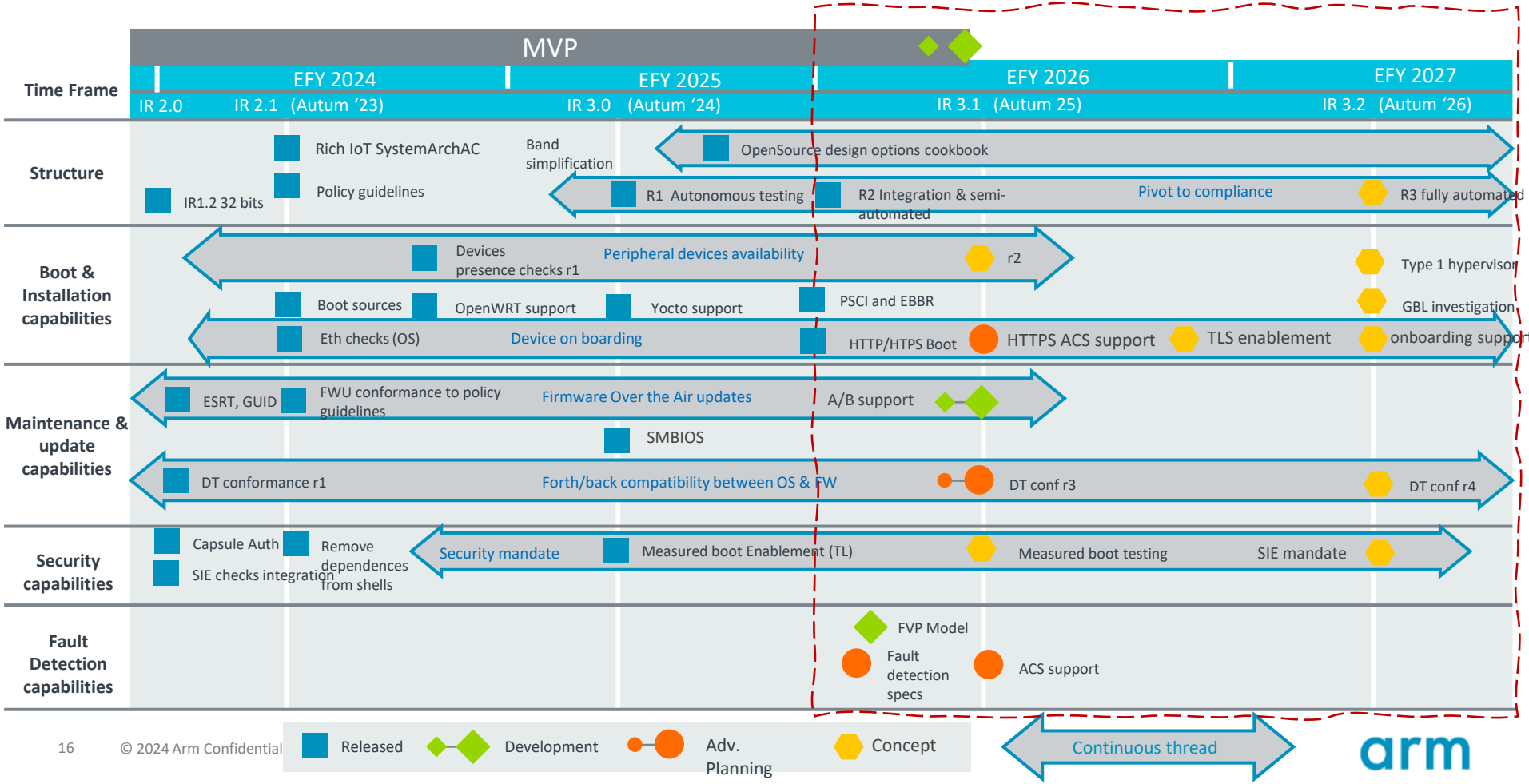
Automotive capabilities

Hardware fault detection support

- Hardware safety may require detection of permanent faults
  - Via hardware or software tests running at EL3 (STLs)
  - During cold boot or periodically at runtime
- Software responsible to
  - Schedule runtime tests
  - Define which test subset to execute on each run
  - ➡ Balance fault detection with operational needs
- New Firmware Interface written by Arm
  - Standard SMC ABI in a new specification
  - Detect firmware support and request test execution
- Confidential version available on request and under NDA
  - Public version at alpha quality in June 2025

Roadmap

SystemReady DT capabilities Roadmap



Q&A

Thank You!