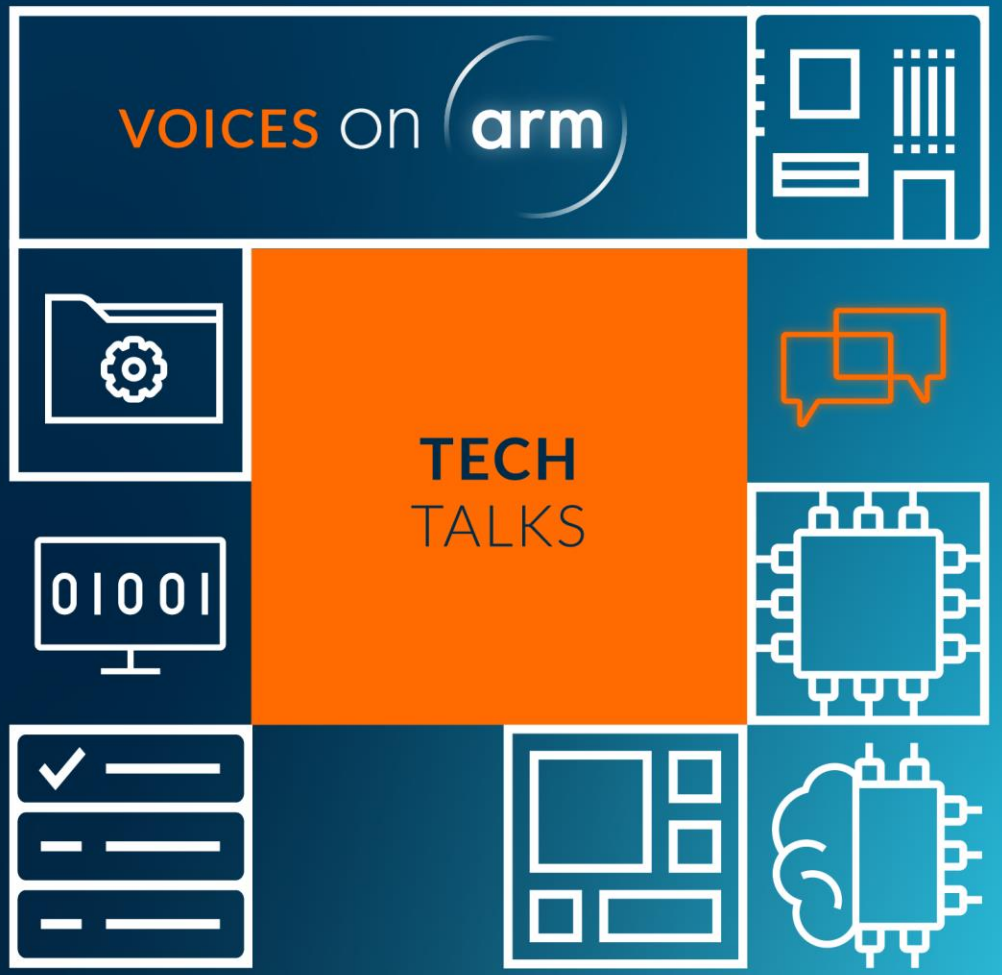
arm

Securing IoT with PARSEC and AWS Greengrass



Guest: Darragh Grealish – 56K.Cloud

21st February 2023

PARSEC

56K CLOUD

Welcome!

Tweet us: #ArmTechTalks

View tech talks on-demand:
www.youtube.com/arm

Sign up for upcoming tech talks:
www.arm.com/techtalks


The Future is Built on arm
Visit Arm at Embedded World
Hall 4, Stand 4 – 504
www.arm.com/embedded-world

Our Upcoming Arm Tech Talks

Date	Title	Host
Feb 21st	Securing IoT with Cloud Native Tooling, PARSEC and AWS Greengrass	56k Cloud
Feb 28th	How to reduce Friction at the Edge and Bootstrap Your IoT Projects	Eurotech
Mar 7th	Fast Development of Noise Detection ML Models: Qeexo AutoML and Arm Virtual Hardware	Qeexo
March 14th: Embedded World		

arm

Speaker: Darragh Grealish

Working at 56K.Cloud as CTO

ARM Ambassador Program,

Docker Captain

twitter.com/grealish

darragh@56k.cloud

Darragh Grealish

CTO – 56K.Cloud

Challenges Currently

Security is difficult and limited use cases doing it properly!

Problems with doing in right

- Ecosystem is broad and scattered
- Limited languages available, mostly limited to C/embedded C
- Siloed into low-level implantation based on specific platform
- Know how is limited and very specific
- Various properties, vendors specific

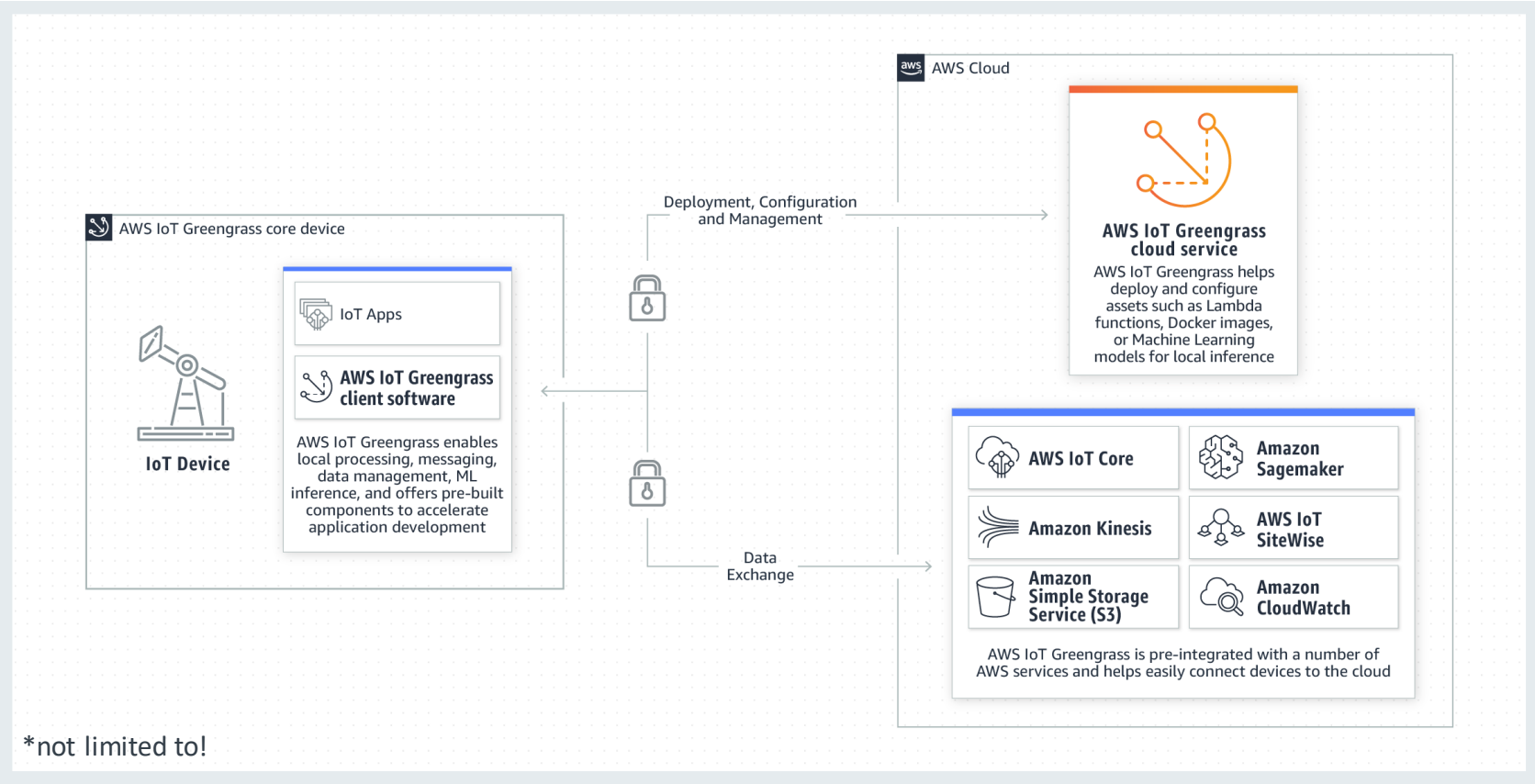
Easy way out (what we see today)

- Securing the Operating system (jail locking)
- RoT is used in limited scope, /dev/tpm the application might own
- Fusing firmware but the OS is unsecure,
- Application owns just TPM or USB HSM*
- Securing point of entry is “good enough”
- Security is someone else's problem

Amazon Web Services – (AWS) Greengrass IoT

What is AWS Greengrass and IoT offering

- A core component and services to support communication and management of both application lifecycle and data on devices*



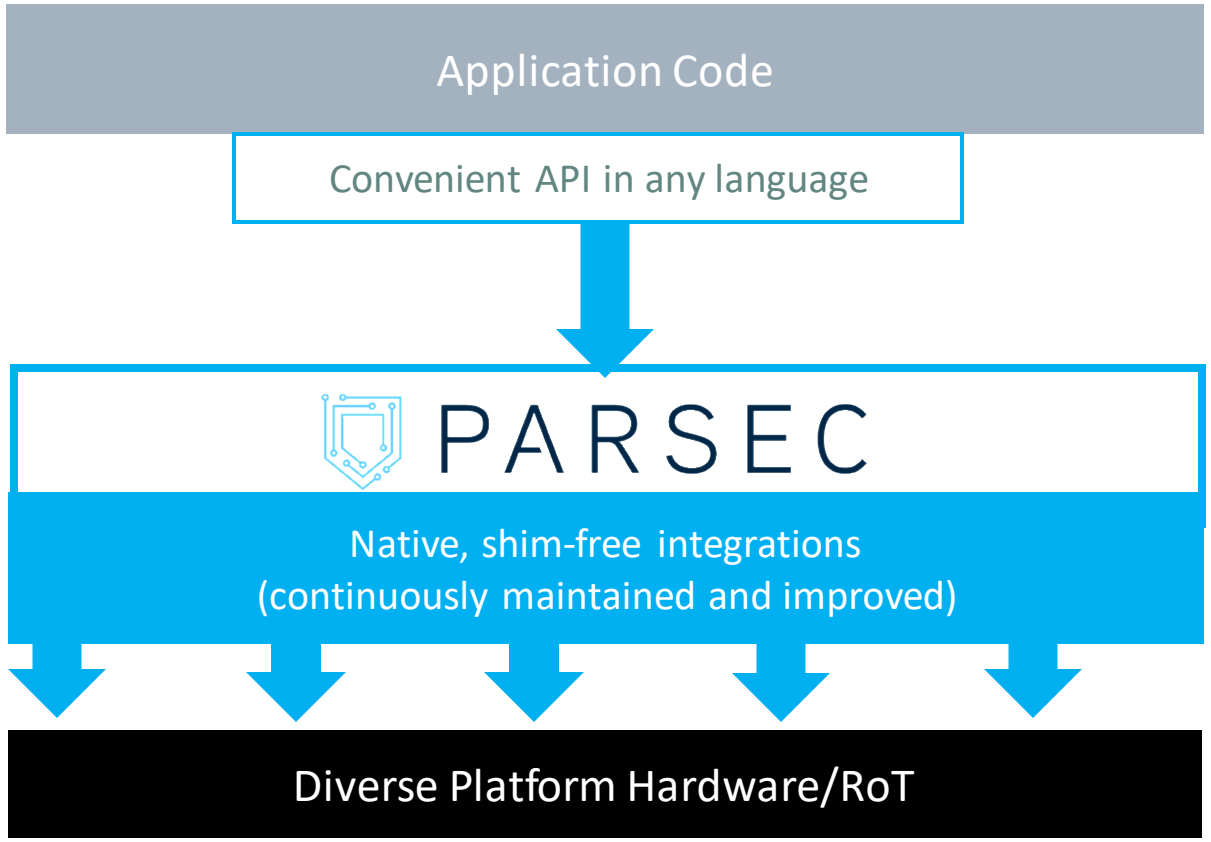
Ref: AWS Website

What is Parsec?

Security Solution for Cloud Native Development at the Edge

arm

How does Parsec add value?



Simplifies security: Parsec exposes cloud native functions to applications and deals with connecting to the Root of Trust (RoT) in your system.

Enables portability: With a common interface for security, you can move your code to any device and connect to the RoT without changing a line of code in your applications.

Isolates applications: Parsec brokers access to hardware and provides isolated key stores.

Accelerates Development: Developed and maintained by a community of security experts who leverage the service in production.

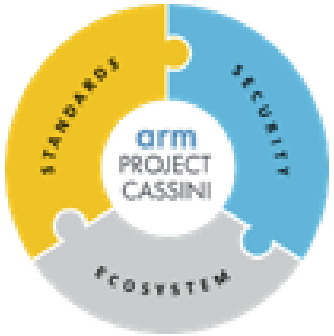
Who is involved in the project?


Scalys
Security at the edge
pelion
NXP
fedora
arm
NETFOUNDRY
SPIN UP YOUR NETWORK
docker
MIRANTIS
aws open source
SolidRun
Embedded Edge Computing
CONFIDENTIAL COMPUTING CONSORTIUM
nixu
cybersecurity.
Red Hat
openSUSE
yocto PROJECT
spiffe
MyAIs Tech
SPIRE
56K CLOUD
CLOUD NATIVE SANDBOX

arm

PARSEC is part of Arm Cassini

Arm is actively engaging with partners to make Parsec the defacto standard for Cloud Native security



- PARSEC — Open API for cross-platform security services
- psacertified™ — Security framework & independent certification
- arm SystemReady — Hardware, firmware specifications Certification program

Cloud Native Stacks

Cassini Edge Solution Reference Implementations

Aligning with SIPs, ODMs and OEMs

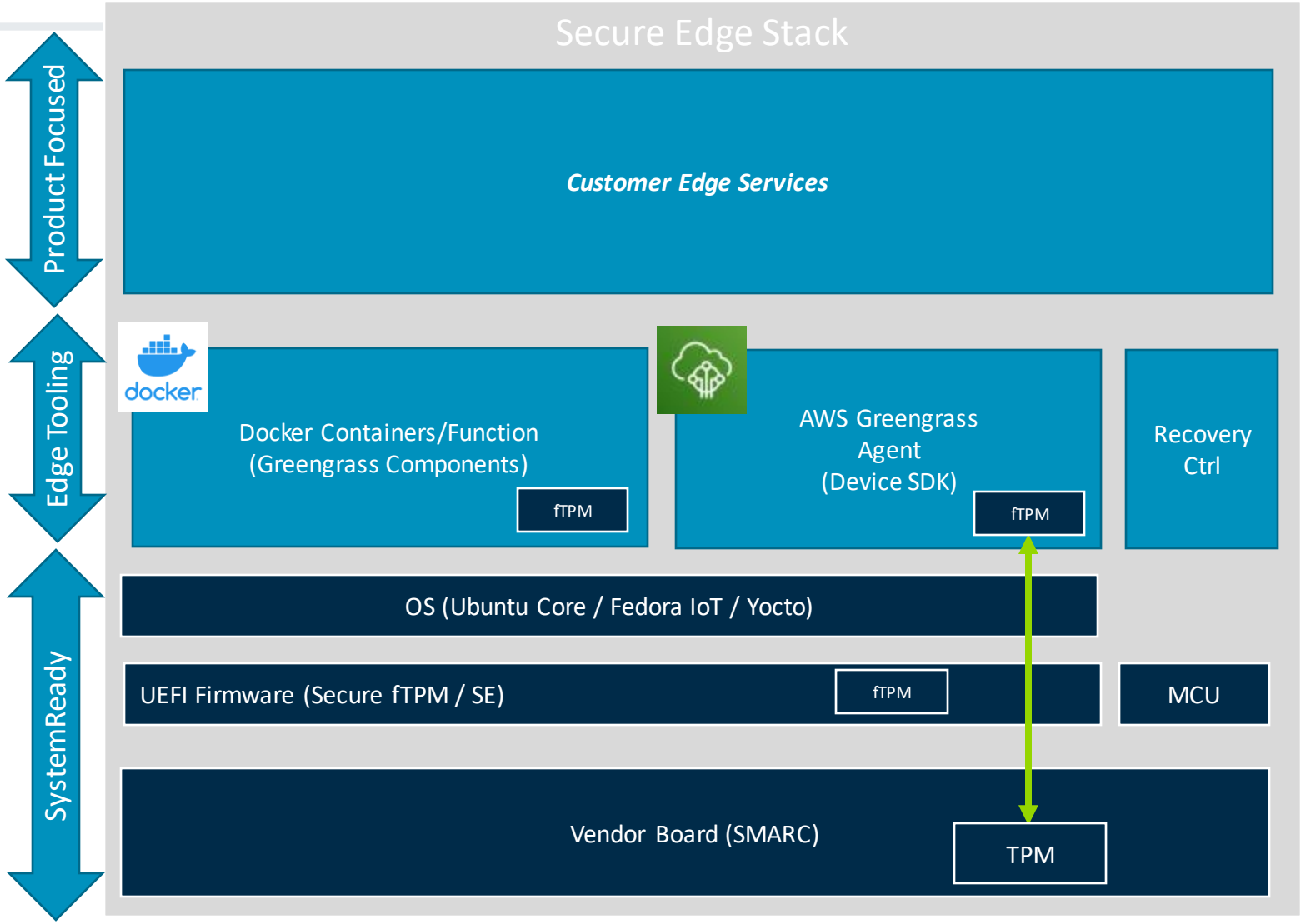
https://www.arm.com/solutions/infrastructure/edge-computing/project-cassini

Securing the Stack

Example of a typical connected software stack on a device

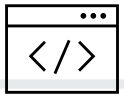
Make Best use of:

- Common HW platform
- Decoupled stack
- Firmware based Trusted Platform Module (fTPM)
- Build on existing ecosystem (UEFI/EDK + SR)
- Multi-tenet secure element
- Secure GG Provisioning



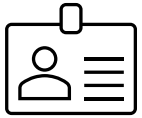
What is available today?

Client Interfaces



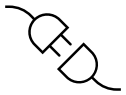
- Rust Client
- C Client (PSA)
- Go Client
- Java Client
- Shell/CLI Tool
- Python Client
- Other Languages
- Enhanced/Smart Clients With Simplified Interfaces

Client Authentication



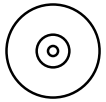
- Peer Credentials
- SPIFFE Identity

Wire Transport



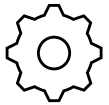
- Unix Domain Socket
- Shared Memory
- TCP Socket

OS Support



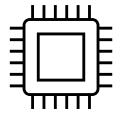
- Linux
- Windows
- RTOS/Specialized

API



- Key Creation
- Key Import/Export
- Sign/Verify Hash
- Asymmetric Encryption
- Hashing
- AEAD
- Random Numbers
- KDF/Key Agreement
- Key Source Attestation
- Sign/Verify Message
- Symmetric Encryption
- HMAC
- Blob Storage

Back-end Providers



- TPM 2.0
- PKCS#11
- Mbed TLS
- CryptoAuthLib
- Trusted Services in OPTEE

Available *Coming Soon* *In Discussion*

How PARSEC supports AWS Greengrass

Providing a simplified path to Hardware based security

Provision keys with Parsec directly in the device HW

Root of Trust (RoT)

Connect devices to GG service securely with certificate exchange.

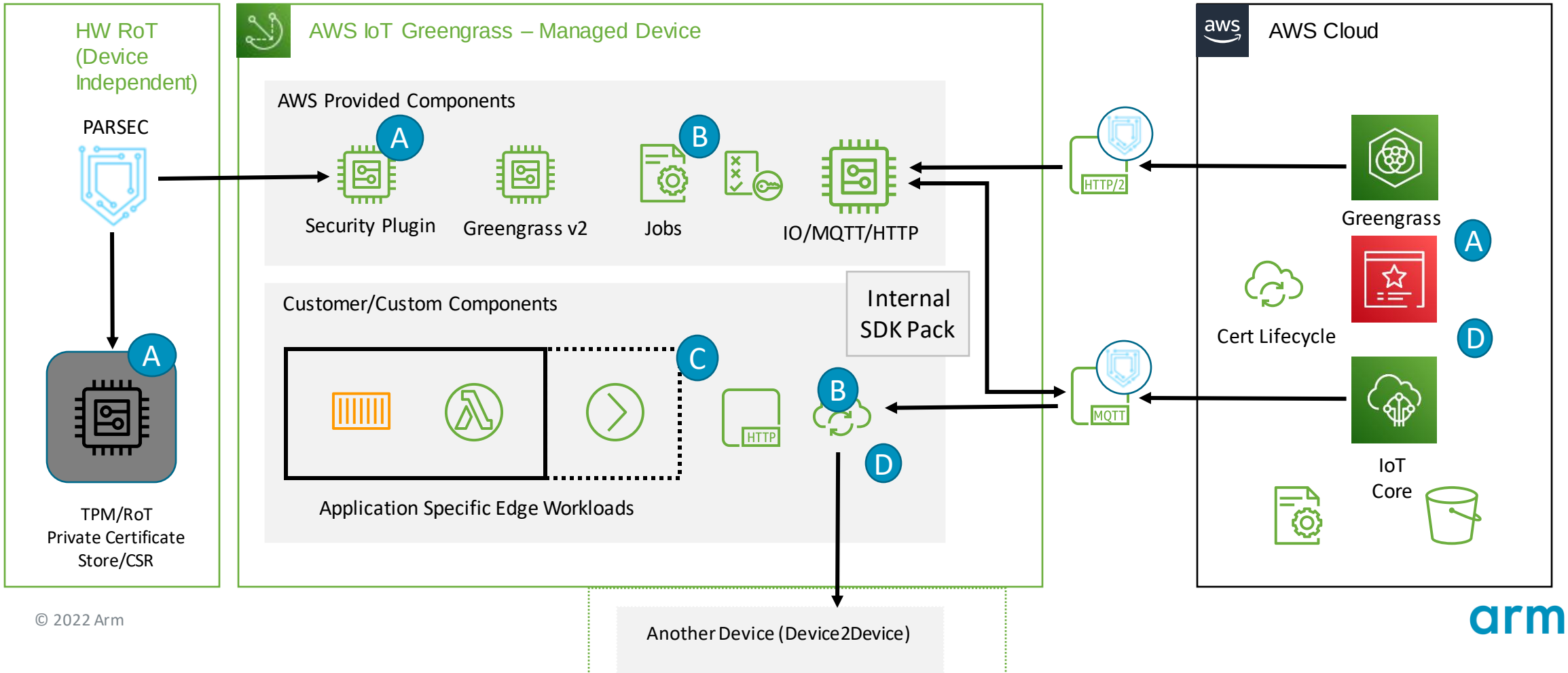
Private key never leaves the RoT

Perform data collection and analysis securely in Edge devices.

arm

Example of Device Deployment

- **Inherits certification, device, lifecycle management from AWS (ACM/ Private CA)**
- **Customer/Custom Components allow (/w GG SDK) to build on MQTT and mTLS HTTP**
- **Can be building into as part of Internal SDK**
- **Uses Cloud-Native Components (PARSEC), (Containers, larger devices)**



arm

Demo Hardware

MACCHIATObin
Double Shot
is now SystemReady ES certified

arm
SystemReady
ES

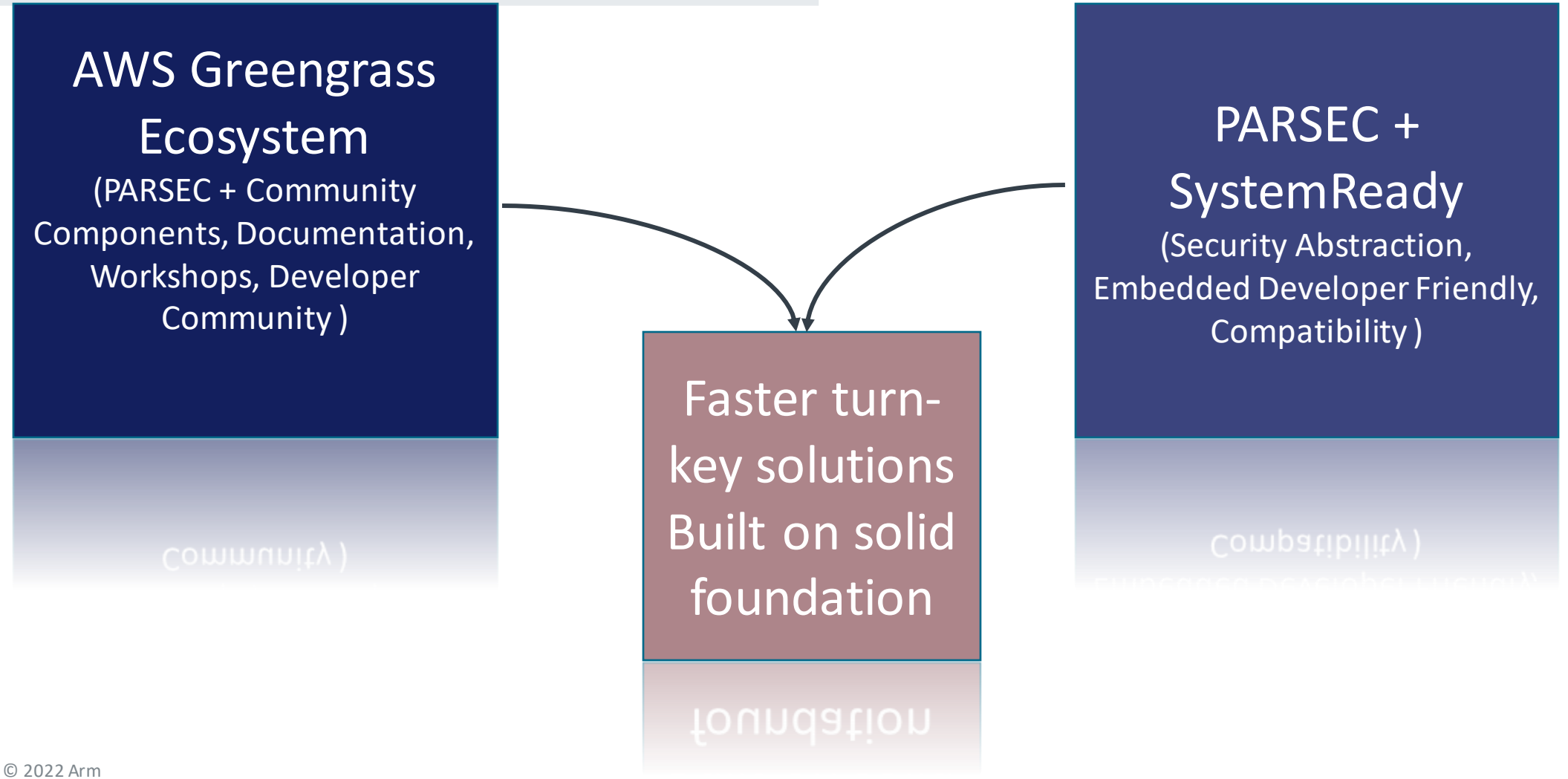
- **Macchiatobin** – Marvel Armada 8040:
- **A Solid-Run SBC focused on Network Applications**
- **ARM SystemReady ES** Certified (July 2021)
- Large community across firmware and OS support (EDKII, UEFI, Fedora)
- Spec: CPU Marvel **ARMADA 8040** Quad core **Cortex-A72,** 16GB RAM,
- Interfaces: Dual x2 10GbE SPF+, 1x 2.5GbE SPF+, 1x 1GbE,
- Storage: microSD Card, eMCC, 3x SATA,
- OS: Fedora 35, OpenWRT, Ubuntu

PARSEC

arm

Two Core Elements

Full Turn-key Program



How do I get started ?

Available as a supported package…

- How to Install

```
[admin@fedora ~] $> sudo dnf install parsec
[admin@fedora ~] $> sudo dnf install parsec-tool
[admin@fedora ~] $> sudo systemctl start parsec
[admin@fedora ~] $> parsec-tool ping
[INFO] Pinging Parsec service...
[SUCCESS] Service wire protocol version is 1.0.
```

- Part of a recipe…

- Or build from source on any Linux.

arm

To find out more:

Greengrass DEMO
https://github.com/56kcloud/parsec-workshop

PARSEC

https://parsec.community/

https://aws.amazon.com/greengrass/

56K CLOUD

https://56k.cloud/

Thank You

Danke
Gracias
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה

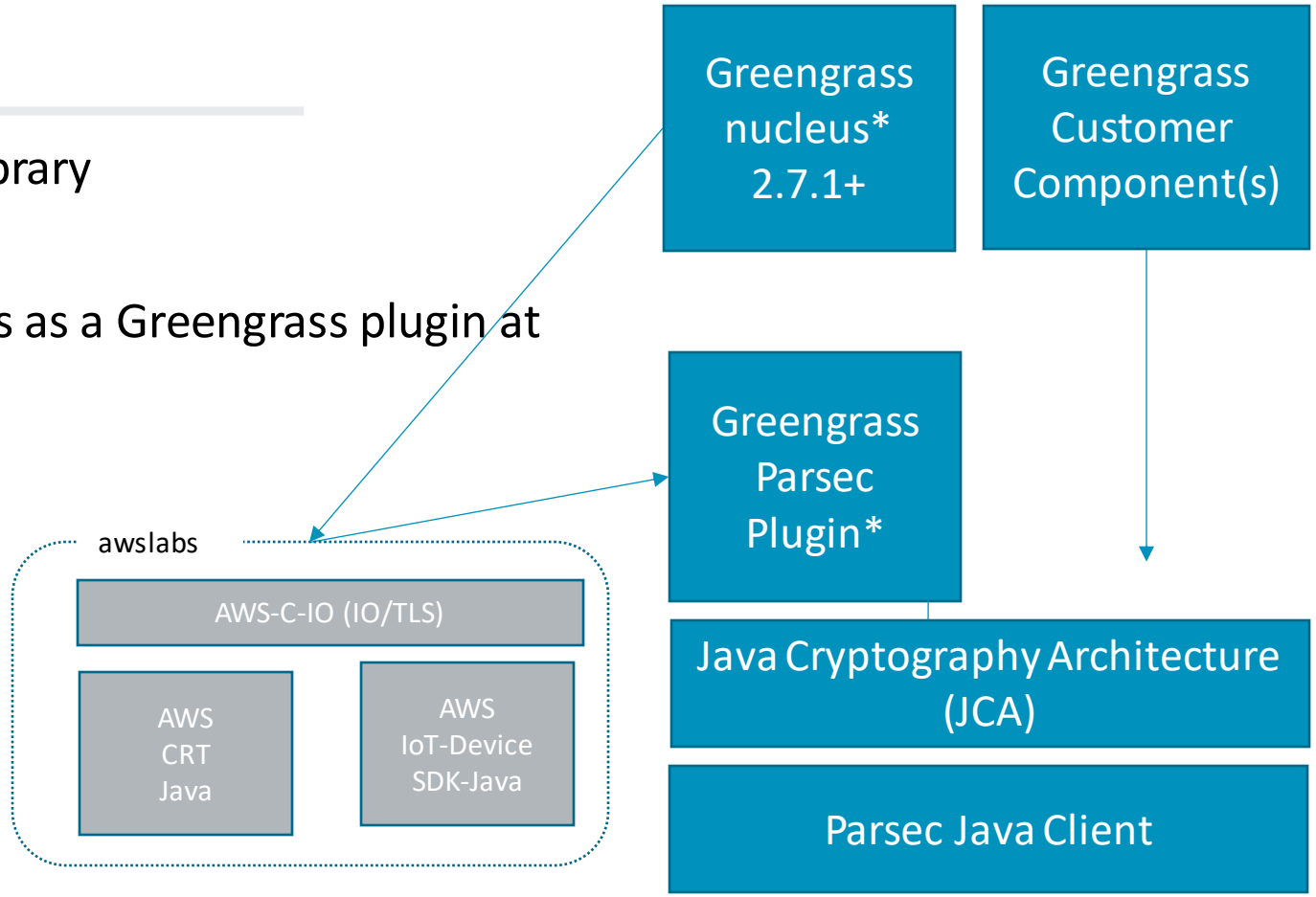
AWS Greengrass

How it's structured

- Built on-top of **parsec-java-client** library
- **parsec-jca-provider** packages this
- **greengrass-parsec-jca-provider** acts as a Greengrass plugin at runtime

Build Steps (in order)

- Build the AWS-C-IO Module
- Build AWS CRT (Common Runtime)
- Build the AWS Device SDK Modules
- Build Parse Java Client build in Maven Central
  - Package and Publish to parsec-jca plugin
  - to Maven Central / Docs / Workshop



Three Branches with pending changes:

- https://github.com/awslabs/aws-c-io/tree/key-op-prototype
- https://github.com/awslabs/aws-crt-java/tree/key-op-prototype
- https://github.com/aws/aws-iot-device-sdk-java-v2/tree/key-op-prototype

20

Potential AWS Device Qualified + SystemReady Cert

Example of some of secured IoT hardware

Nvidia Jetson TX2
Nvidia Jetson Nano

NVIDIA JETSON TX2

NXP LX2160 /
Marvel 8040

SolidRun Embedded Edge Computing

NXP IMX8 x8 core
4GB 8GB
1~5TB SSD

EUROTECH

NXP IMX.8 / RB5 / Rock
2GB``8GB
1~5TB SSD

ADLINK Leading EDGE COMPUTING

SGeT STANDARDIZATION GROUP FOR EMBEDDED TECHNOLOGIES

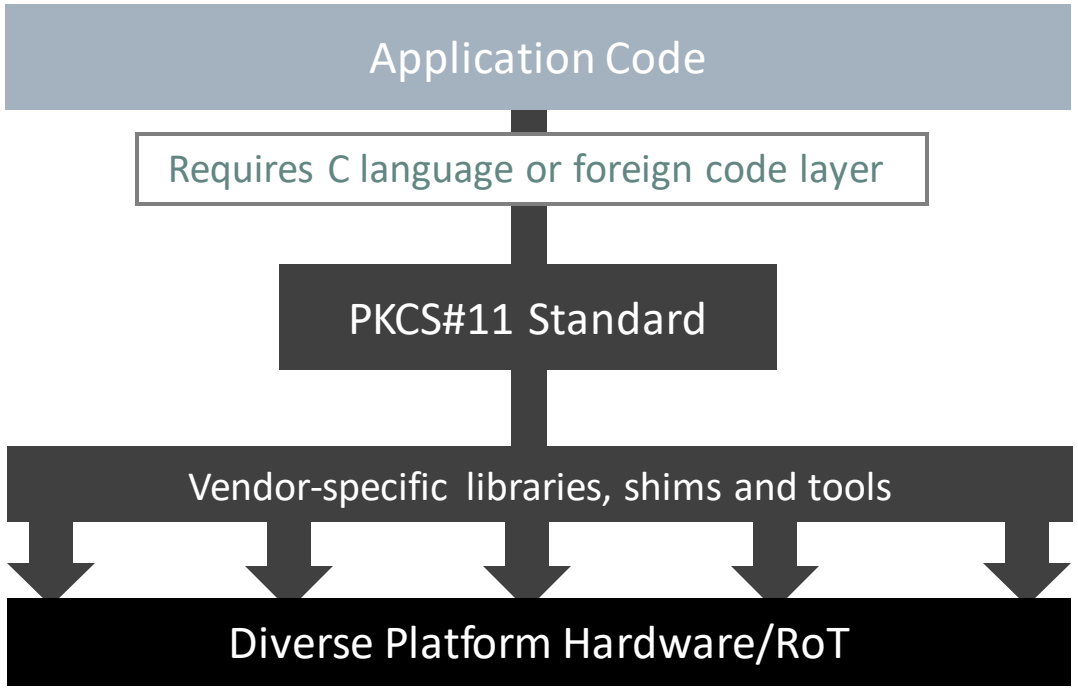
SMARC module

arm

ADLINK Leading EDGE COMPUTING

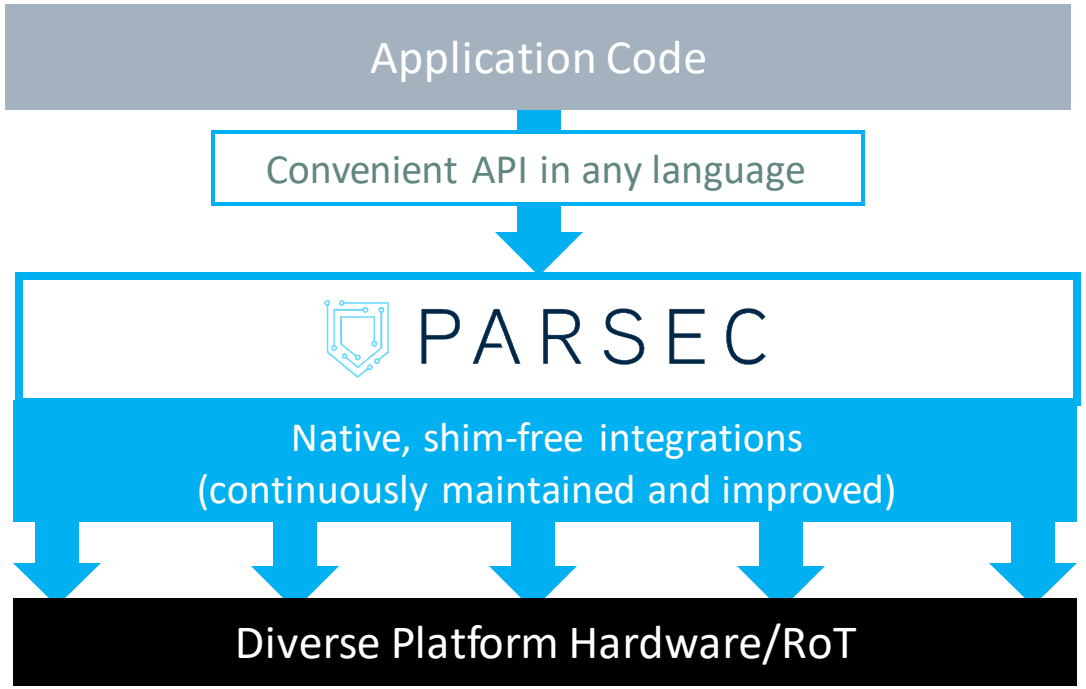
How Parsec Simplifies Hardware Security

Offloading complexity and decoupling from a fragmented ecosystem



The Legacy World

The historical prevalence of the PKCS#11 standard has led to a variety of libraries and shims. Application code needs to directly link and be tested against various libraries on different platforms.



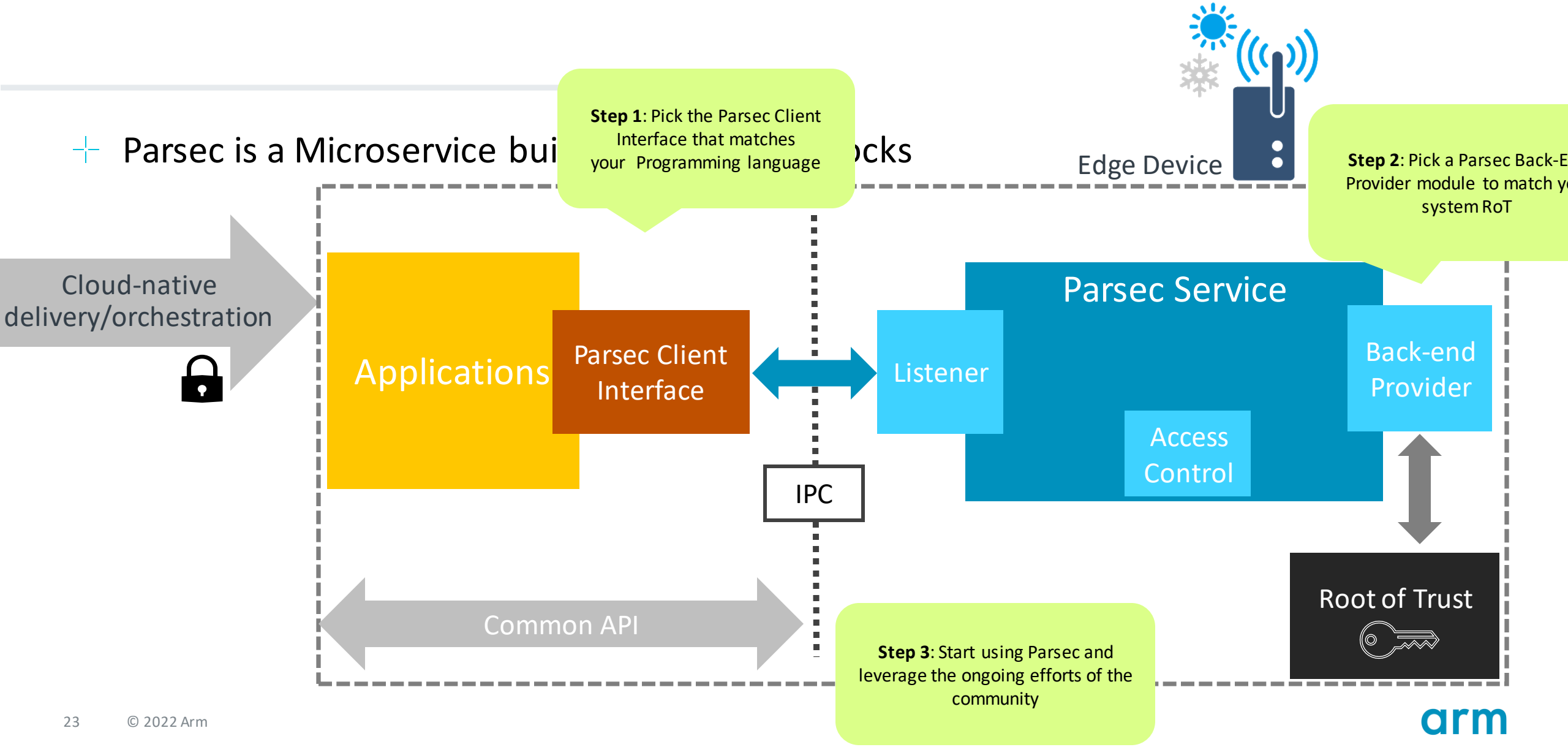
The Parsec World

The Parsec microservice runs as part of the platform, meaning that application code has only a single component to interact with. Parsec handles the variety of integrations needed for different platforms and is maintained by an expert community.

arm

How does it work?

- Parsec is a Microservice bui ocks



Demo

Summary:

- **Greengrass** Installs and Provisions (register as “Thing”)
- Keys are Provisioned and Private Key stored in Parsec (External Workflow)
- Observe: Parsec security agnostic interface is running, Greengrass Service
- In the **AWS Console** we see the device (“Thing”), and trigger a deployment
- **Observe**: Greengrass Nucleus will use the private keys in the security backend provided by Parsec: **MQTT** and **HttpClient** We see in the logs and the Web Console
- Show Parsec service running and log output

Why Parsec?

Cloud

- Rich Workloads
- Multi-Tenant
- Cloud-Native Development

docker

kubernetes

Edge

Endpoint

- Increased Threat Landscape
- Variety of Platforms
- Device-specific RoT

TPM HSM SE

Rich Edge Computing

Occupies the middle ground in between two engineering paradigms: cloud and endpoint (embedded). For rich edge applications, a cloud-native development experience is desirable, but the diversity of platforms means that security is a challenge.

PARSEC Client Library

Github Repo consists of the following:

- Java Client Library
- Java Protobuf Interface (Sources the Parsec Operations repo)
- Java JCA (Java Cryptography Architecture)
- Parsec Java test client implementation (TestContainers)
- Maven Build script
- Build pipeline using Github actions

Location: https://github.com/parallaxsecond/parsec-client-java

Notes – core content of slides / objectives

- Why Security? and even validated security for that matter, (PSA Certified)
- What’s the problem today, or the non-existent of proper security , (look at AWS Greengrass) / this stuff is hard! And no one is talking about, how to do it!
- Lets look at how that's been solved in the Project Cassini – SystemReady Program with PSA certified expectation
- PARSEC: what is it, how does it help in this context
- What is AWS IoT Greengrass, how does Security help in this case, what's required? ?
- How to get started, where to go, what's the plan (CTA / Actions to address in EW32)
- Meet us at Embedded World 2023 (56K.Cloud / AWS / ARM)