PARSEC

PARSEC – Platform Abstraction for Security

Overview

April 2021

Marc Meunier marc.meunier@arm.com

SW Ecosystem Development Team

Paul Howard paul.howard@arm.com

Principle System Solution Architect

PARSEC: An opensource collaborative project

PARSEC

Why Parsec?

Cloud

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

docker

kubernetes

Edge

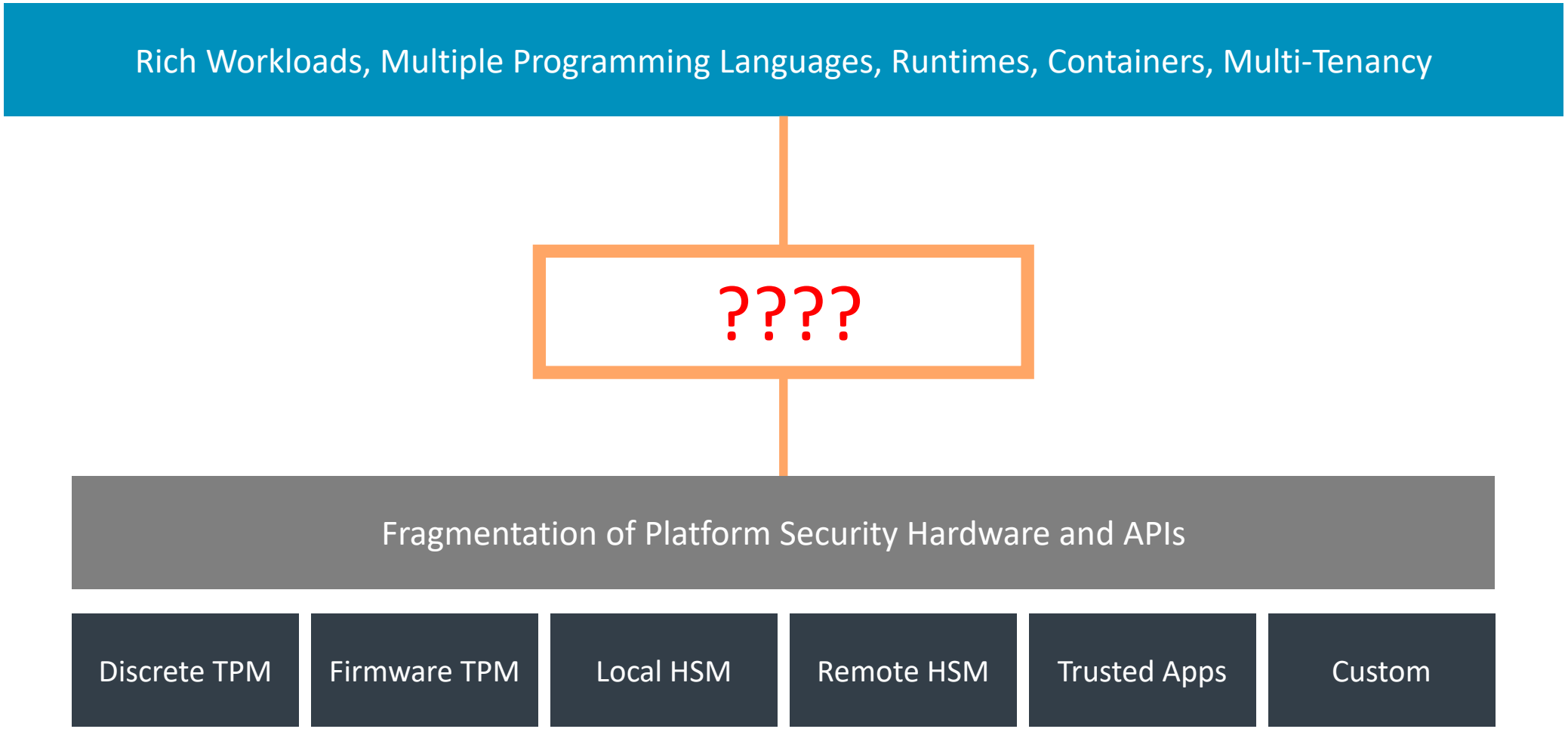
Endpoint

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

TPM HSM SE

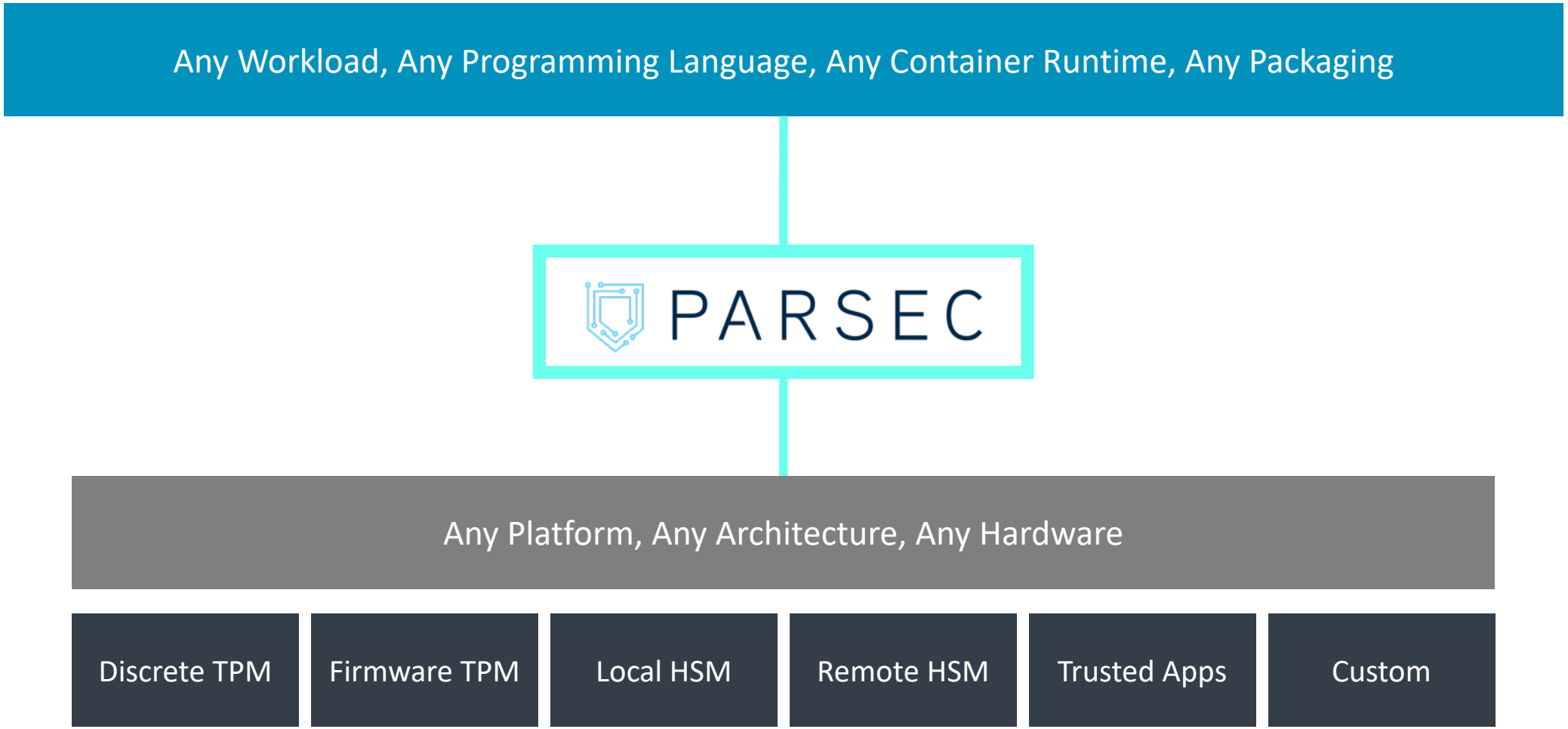
PARSEC

Edge as a Rich Compute Platform – Fragmentation Challenges



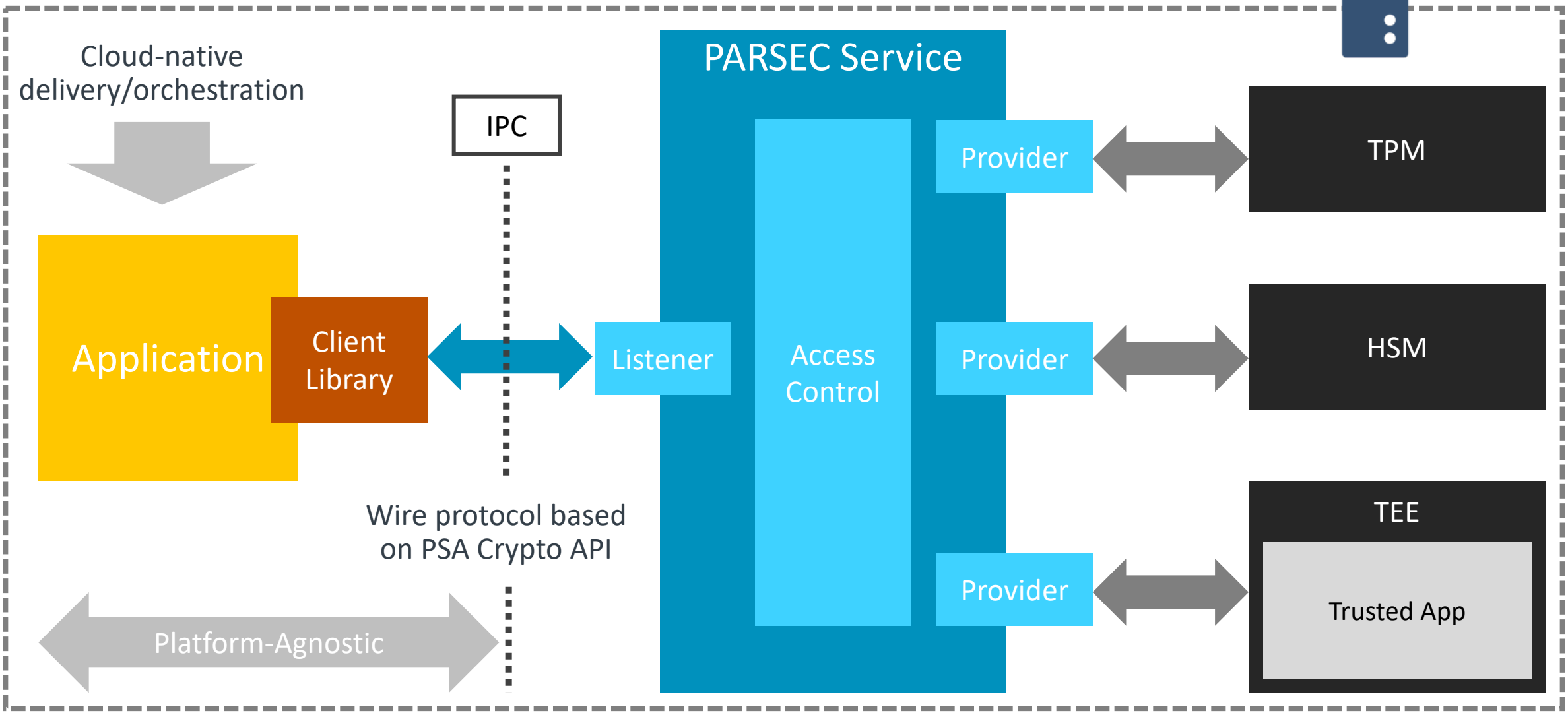
PARSEC

PARSEC: A Platform Abstraction For Security



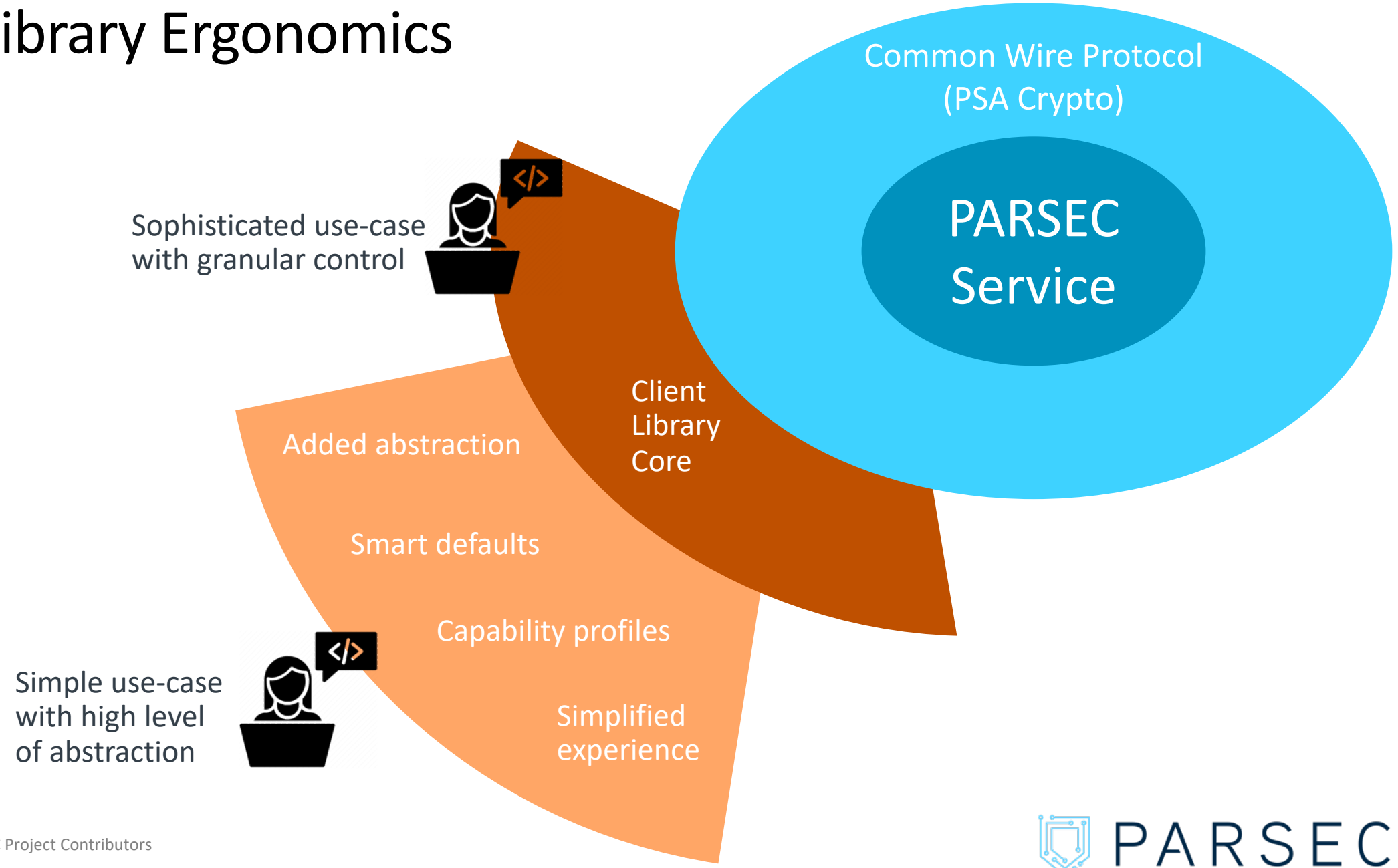
PARSEC

Service Architecture



PARSEC

Client Library Ergonomics

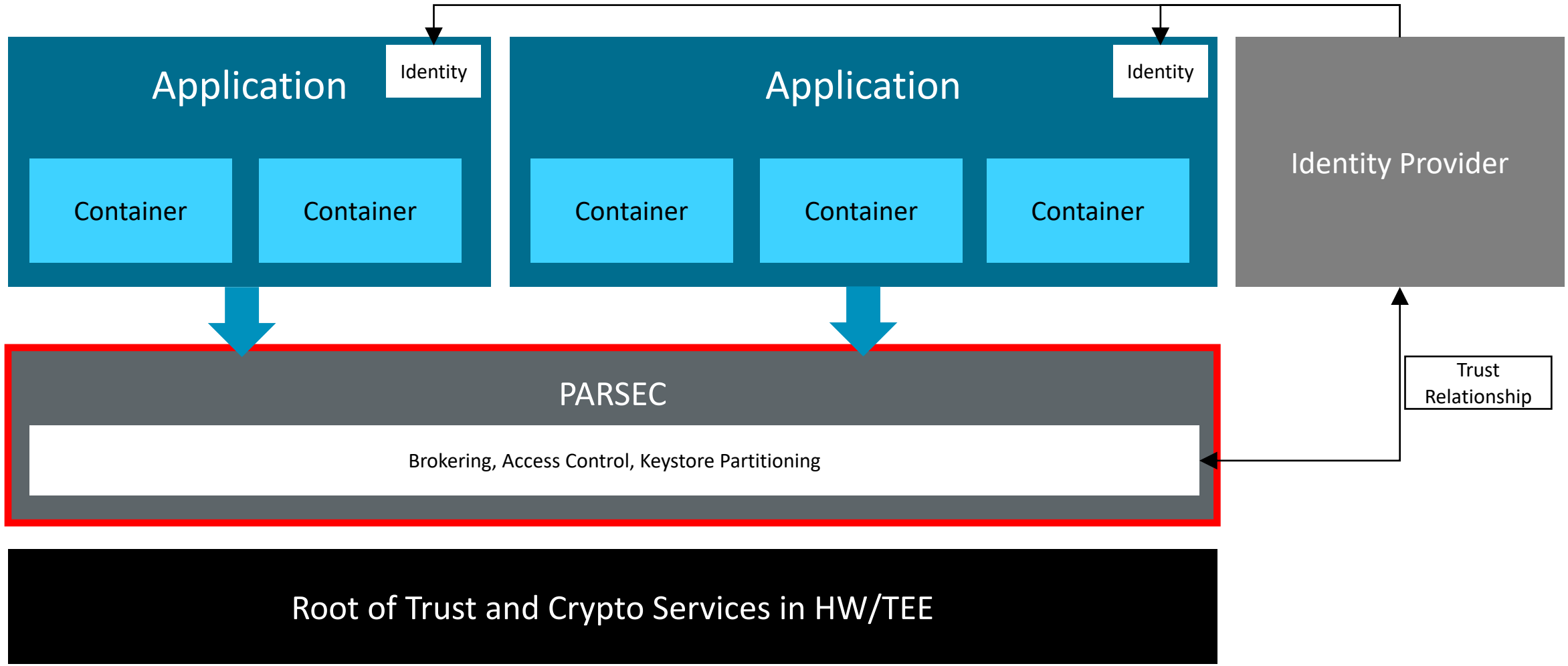


PARSEC

Multi-tenancy

- Cloud-native workload delivery at the edge is expected to be **multi-tenant**
  - Multiple applications will **share access** to the secure hardware on the platform
  - Each application needs a **unique and persistent identity**, which is assigned by a component that PARSEC can trust
- The PARSEC service:
  - Treats each API call according to the identity of the application that made the call
  - **Partitions key stores and brokers access** to hardware based on the identity
- Identity:
  - The source of identity depends on the deployment – it might be a container runtime manager, for example
  - A deployed system requires an **identity provider** – a separate component that assigns identities to applications, which has a trust relationship with the PARSEC service

Application Identity and Multi-Tenancy



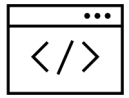
PARSEC

PARSEC Value Proposition

- **Abstraction** – a common API that is truly agnostic and based on modern cryptographic principles
- **Mediation** – security as a microservice, brokering access to the hardware and providing isolated key stores in a multi-tenant environment
- **Ergonomics** – a client library ecosystem that brings the API to the fingertips of developers in any programming language: “easy to consume, hard to get wrong”
- **Openness** – an open-source project inviting contributions to enhance the ecosystem both within the service and among its client libraries

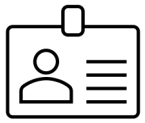
Functional Support Map

Front-End



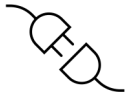
- Rust Client
- C Client (PSA)
- Go Client
- Shell/CLI Tool
- Python Client
- Java Client
- C Client (PKCS#11)
- Other Languages
- Enhanced/Smart Clients With Simplified Interfaces

Client Authentication



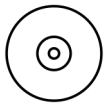
- Peer Credentials
- SPIFFE Identity

Wire Transport



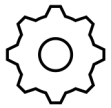
- Unix Domain Socket
- Shared Memory
- TCP Socket

OS Support



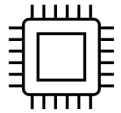
- Linux
- Windows
- RTOS/Specialized

API



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

Back-End



- TPM 2.0
- PKCS#11
- Mbed Crypto
- CryptoAuthLib
- PSA Services in OPTEE

Available *Incomplete or partial support* *Not yet implemented*

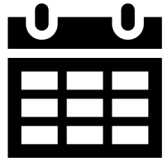
PARSEC

References

https://github.com/parallaxsecond

https://parallaxsecond.github.io/parsec-book

#parsec on CNCF https://cloud-native.slack.com



Weekly community call (see GitHub)

Note: "**parsec**" was already being used as an organization name in GitHub, which is why the expanded "**parallaxsecond**" term was selected instead.

PARSEC