IPADS
INSTITUTE OF PARALLEL
AND DISTRIBUTED SYSTEMS

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Distributed Functionality of OpenHarmony

Dong Du

Shanghai Jiao Tong University

(https://dongd.info/)

Distributed OS: Use Cases

- Use Pad to draw figs

- Play games with bigger screen

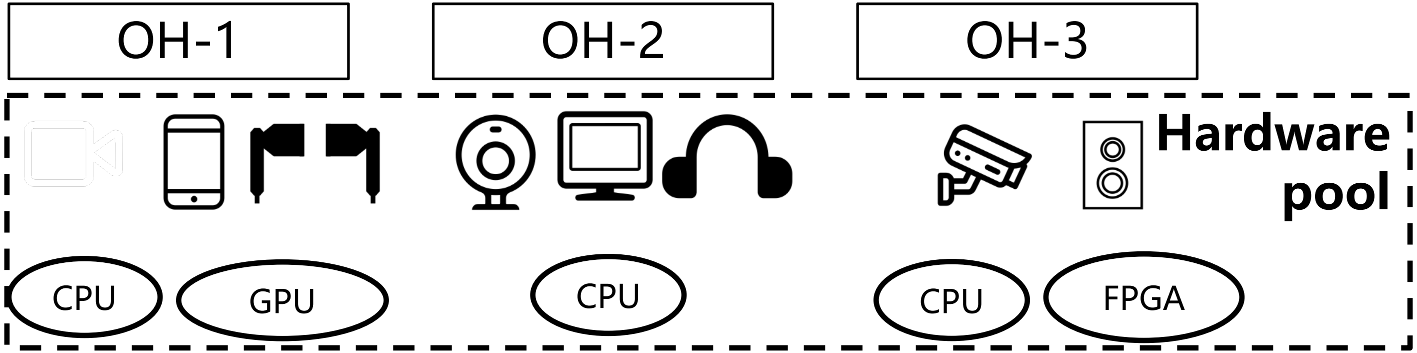
Outline of Distributed OS

- **Distributed hardware/devices**
- **Distributed software bus (D-softbus)**
- **Distributed data**
- **Distributed security**

DISTRIBUTED HARDWARE

Distributed Hardware

- **Multi devices can share their hardware**
  - E.g., one device can directly access remote GPU for computation
- **OpenHarmony provides system supports for distributed hardware**



Distributed Hardware in OpenHarmony

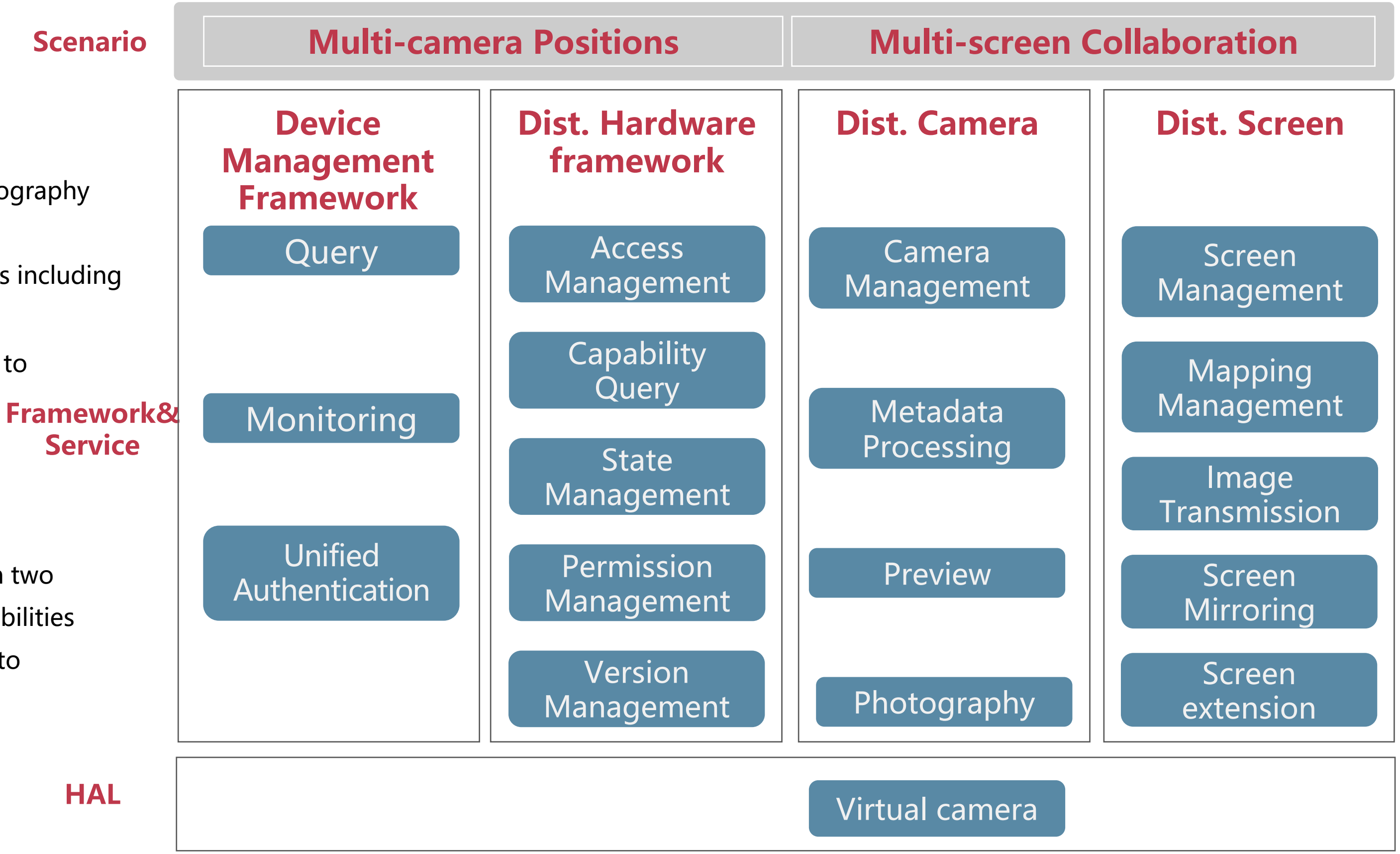
Distributed camera

- Remote camera preview and photography support
- Remote camera control capabilities including zoom and shooting angle
- Performance Metrics: Supports up to 1080P@30fps

Distributed screen

Supports screen mirroring between two devices, enhancing projection capabilities

Performance Metrics: Supports up to 1080P@30fps



Scenario

Multi-camera Positions

Multi-screen Collaboration

Framework& Service

Device Management Framework	Dist. Hardware framework	Dist. Camera	Dist. Screen
Query	Access Management	Camera Management	Screen Management
Monitoring	Capability Query	Metadata Processing	Mapping Management
Unified Authentication	State Management	Preview	Image Transmission
	Permission Management	Photography	Screen Mirroring
	Version Management		Screen extension

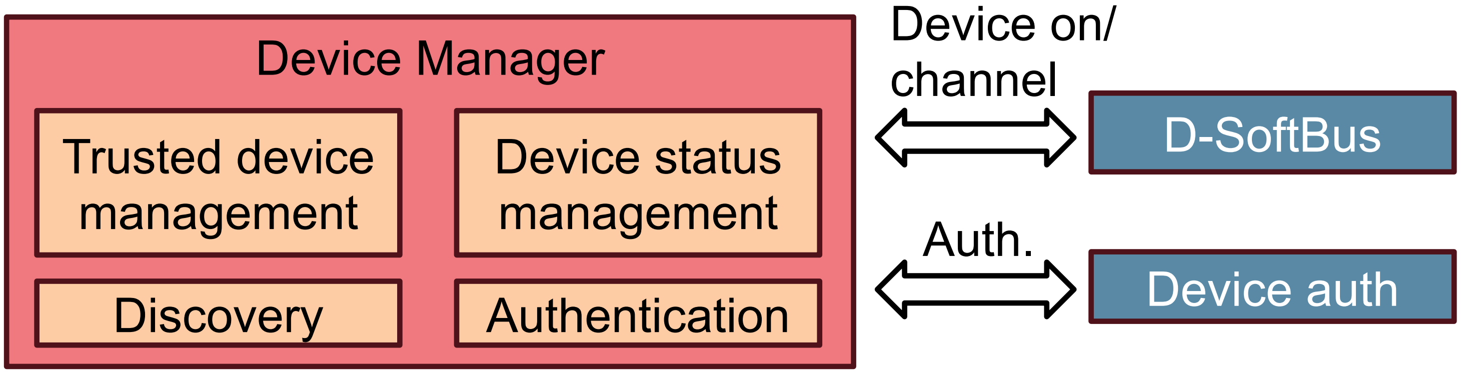
HAL

Virtual camera

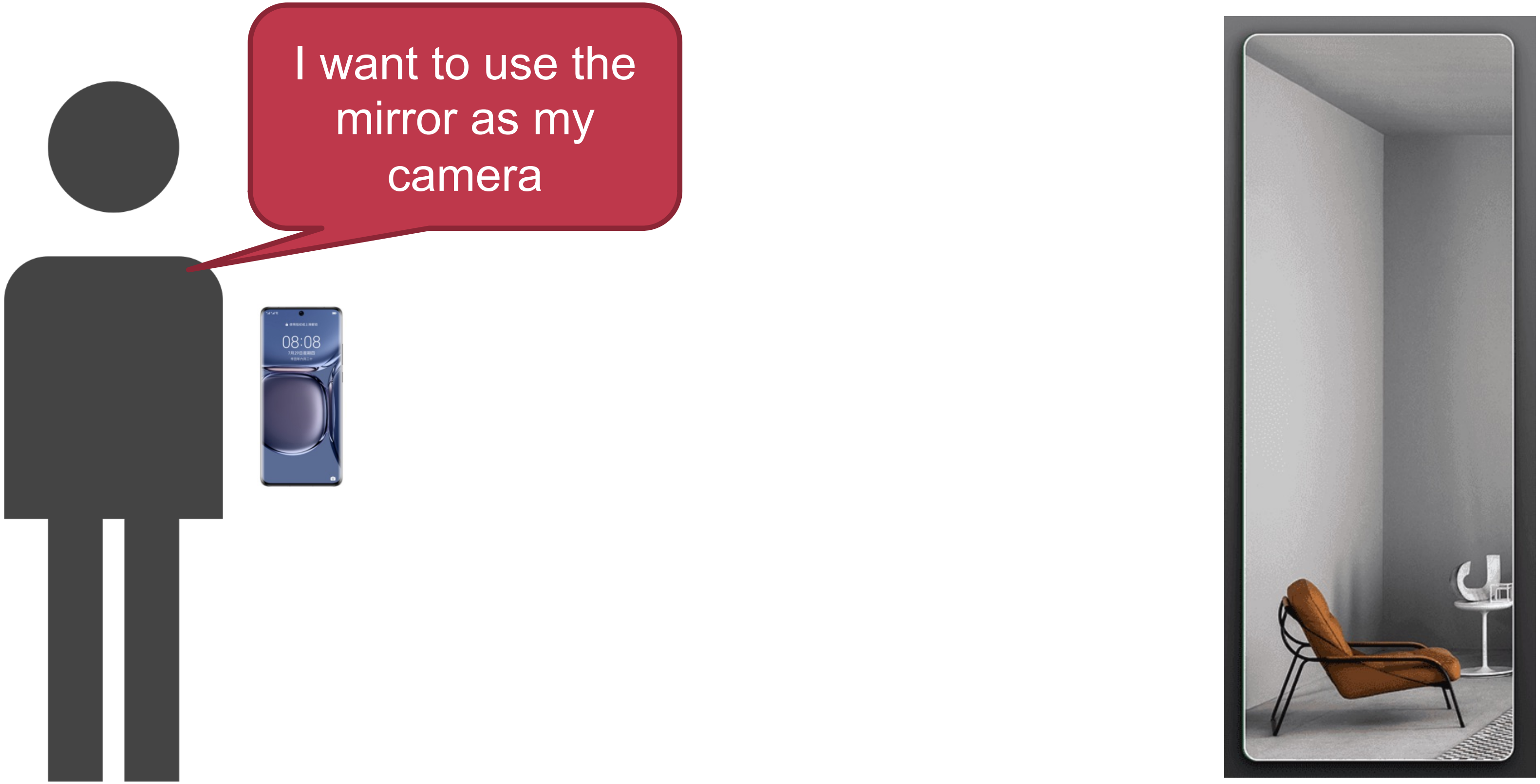
Note: Perf metrics have restriction

Distributed Hardware: Device Manager

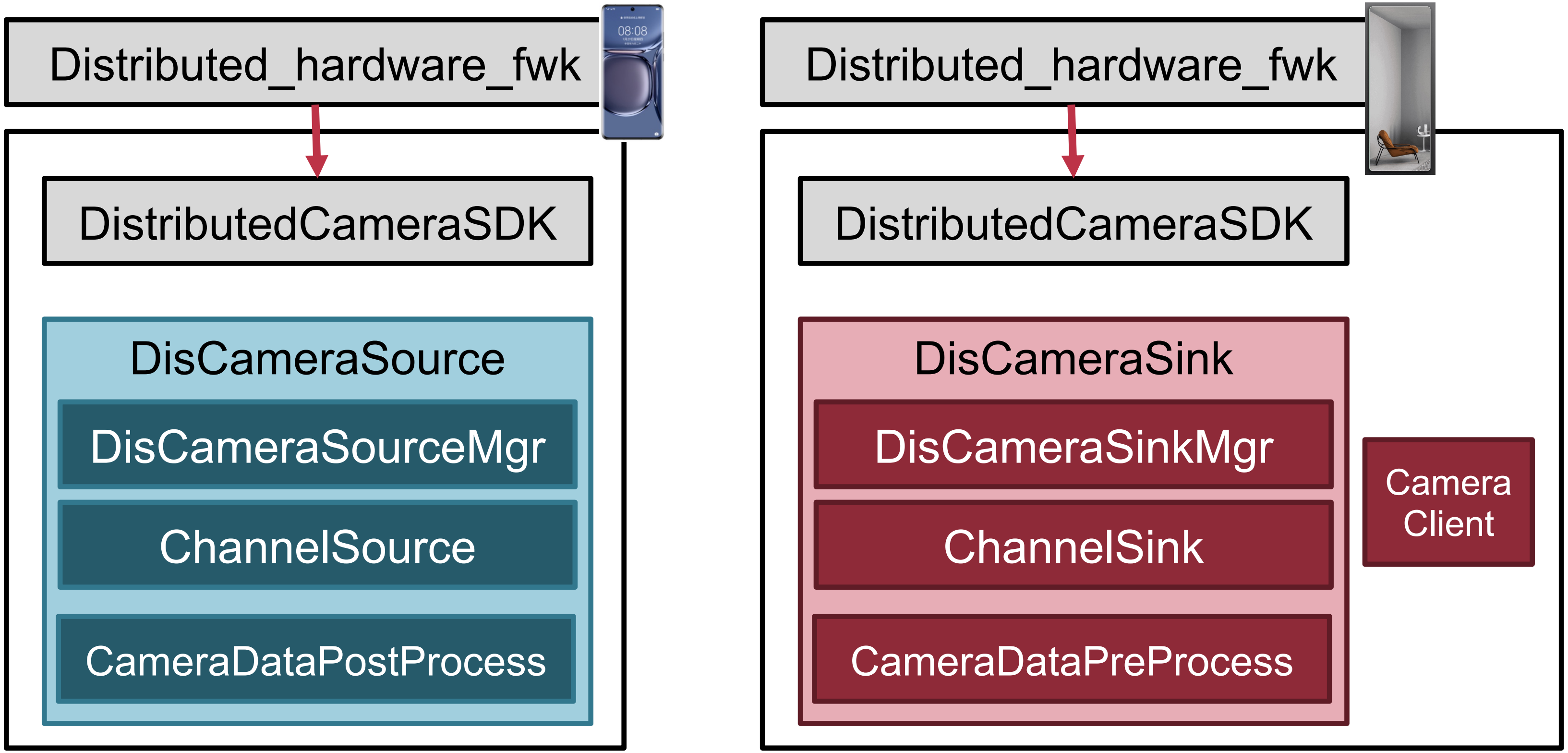
- **DeviceManager supports authentication and networking for account-independent distributed devices**
- **It provides a set of APIs for listening, discovery, and authentication of distributed devices**



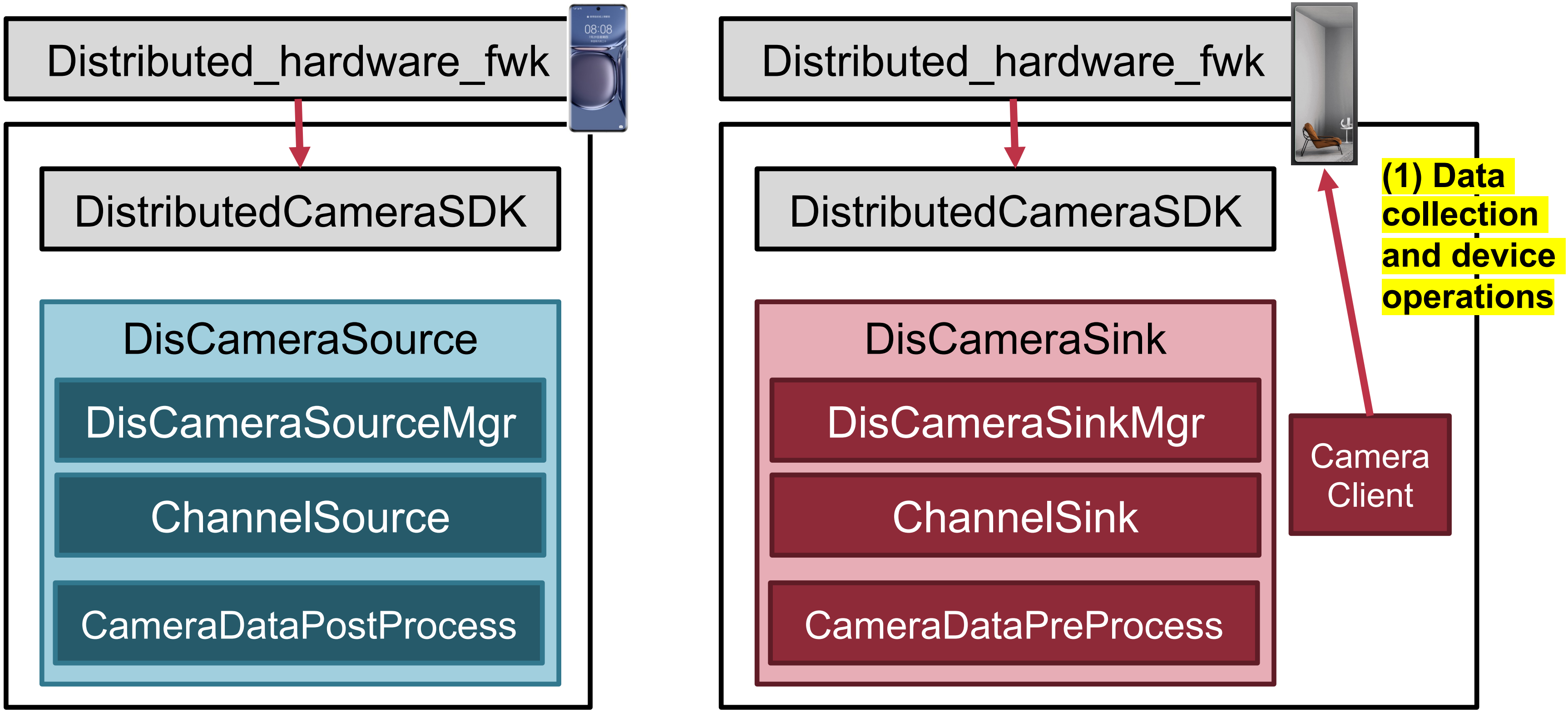
Distributed Hardware: Distributed Camera



Distributed Hardware: Distributed Camera



Distributed Hardware: Distributed Camera



Distributed_hardware_fwk

DistributedCameraSDK

DisCameraSource

DisCameraSourceMgr

ChannelSource

CameraDataPostProcess

Distributed_hardware_fwk

DistributedCameraSDK

DisCameraSink

DisCameraSinkMgr

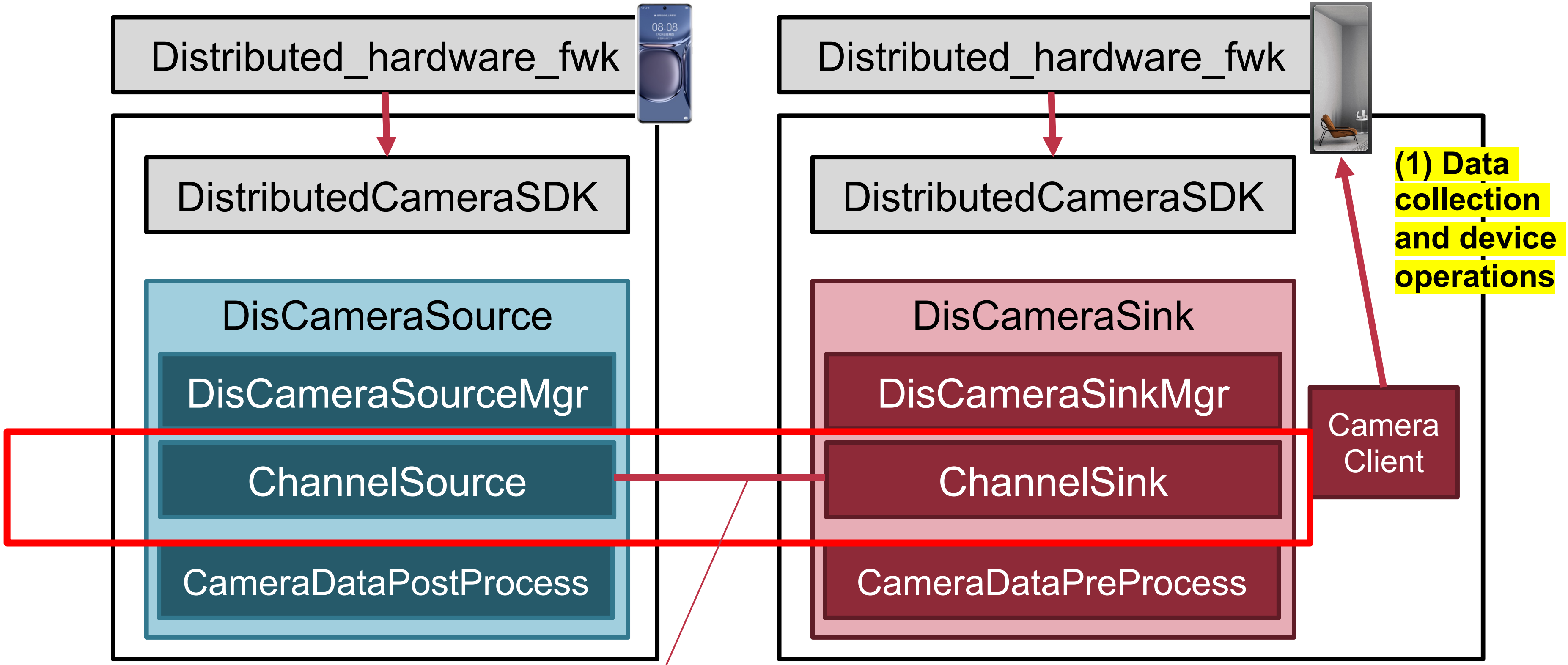
ChannelSink

CameraDataPreProcess

Camera Client

(1) Data collection and device operations

Distributed Hardware: Distributed Camera



Distributed_hardware_fwk

DistributedCameraSDK

DisCameraSource

DisCameraSourceMgr

ChannelSource

CameraDataPostProcess

Distributed_hardware_fwk

DistributedCameraSDK

DisCameraSink

DisCameraSinkMgr

ChannelSink

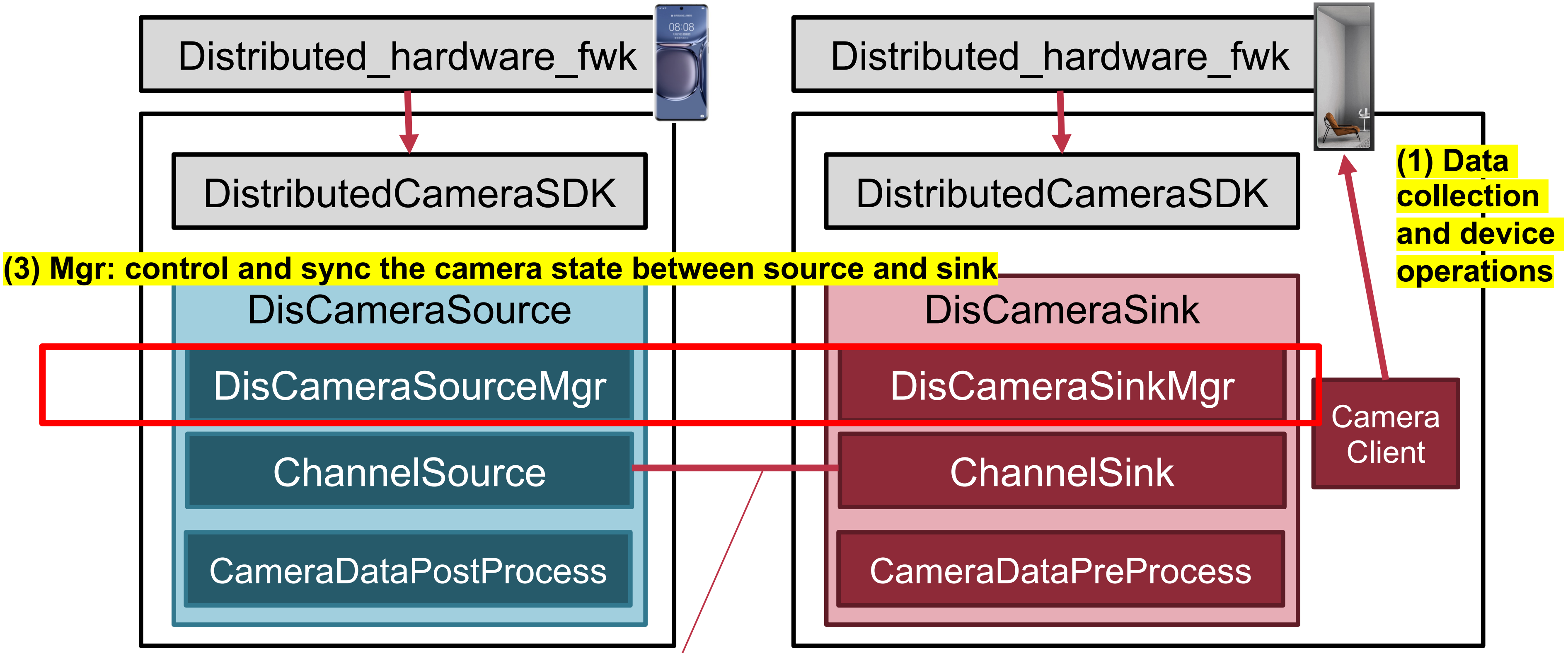
CameraDataPreProcess

Camera Client

(1) Data collection and device operations

(2) Channel: software-bus based channel, exchange both data and metadata

Distributed Hardware: Distributed Camera

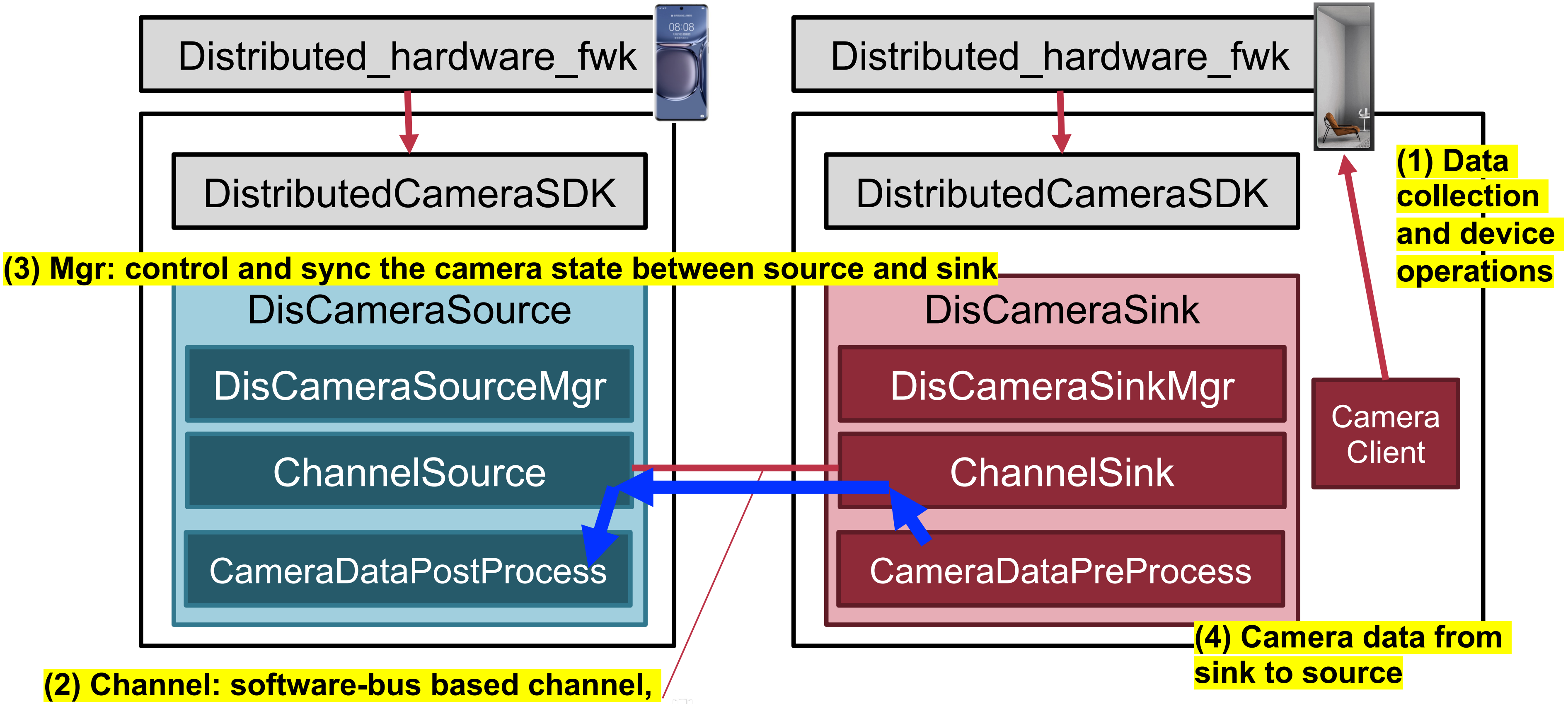


(1) Data collection and device operations

(3) Mgr: control and sync the camera state between source and sink

(2) Channel: software-bus based channel, exchange both data and metadata

Distributed Hardware: Distributed Camera



(1) Data collection and device operations

(3) Mgr: control and sync the camera state between source and sink

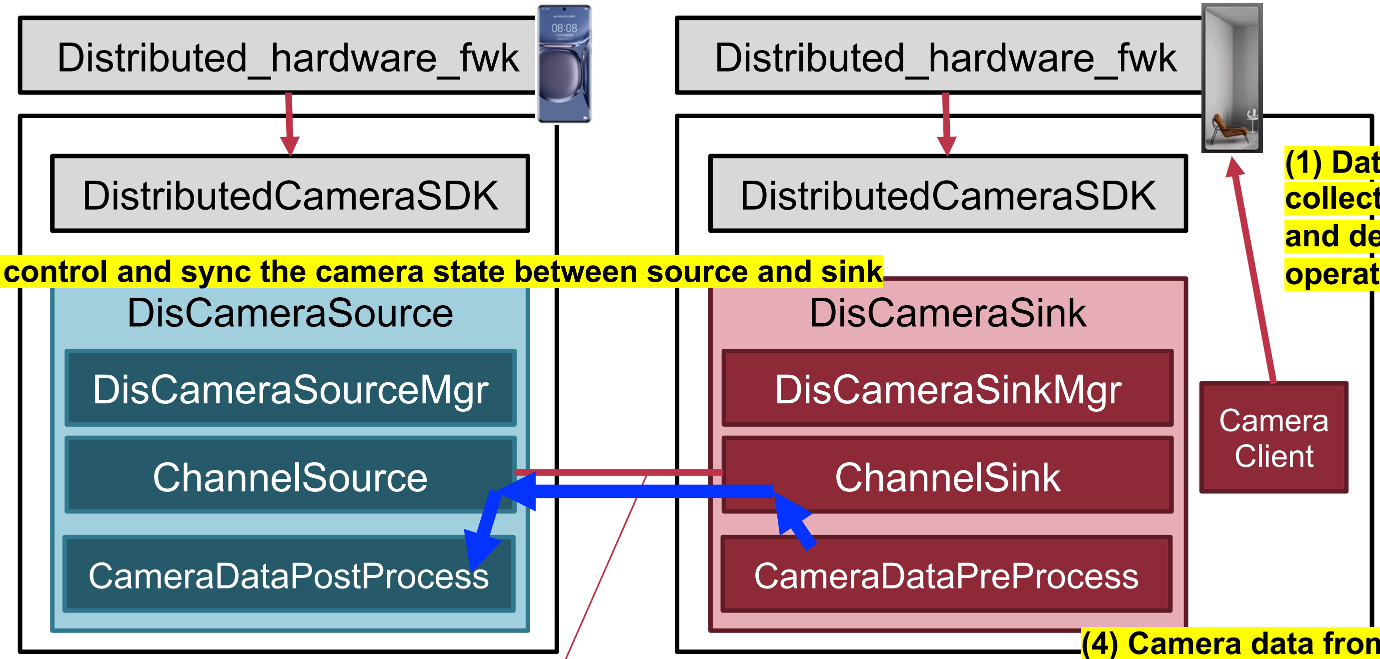
(4) Camera data from sink to source

(2) Channel: software-bus based channel, exchange both data and metadata

Distributed Hardware: Distributed Camera

How can my Apps use the remote camera just like local ones?

Distributed Hardware: Distributed Camera



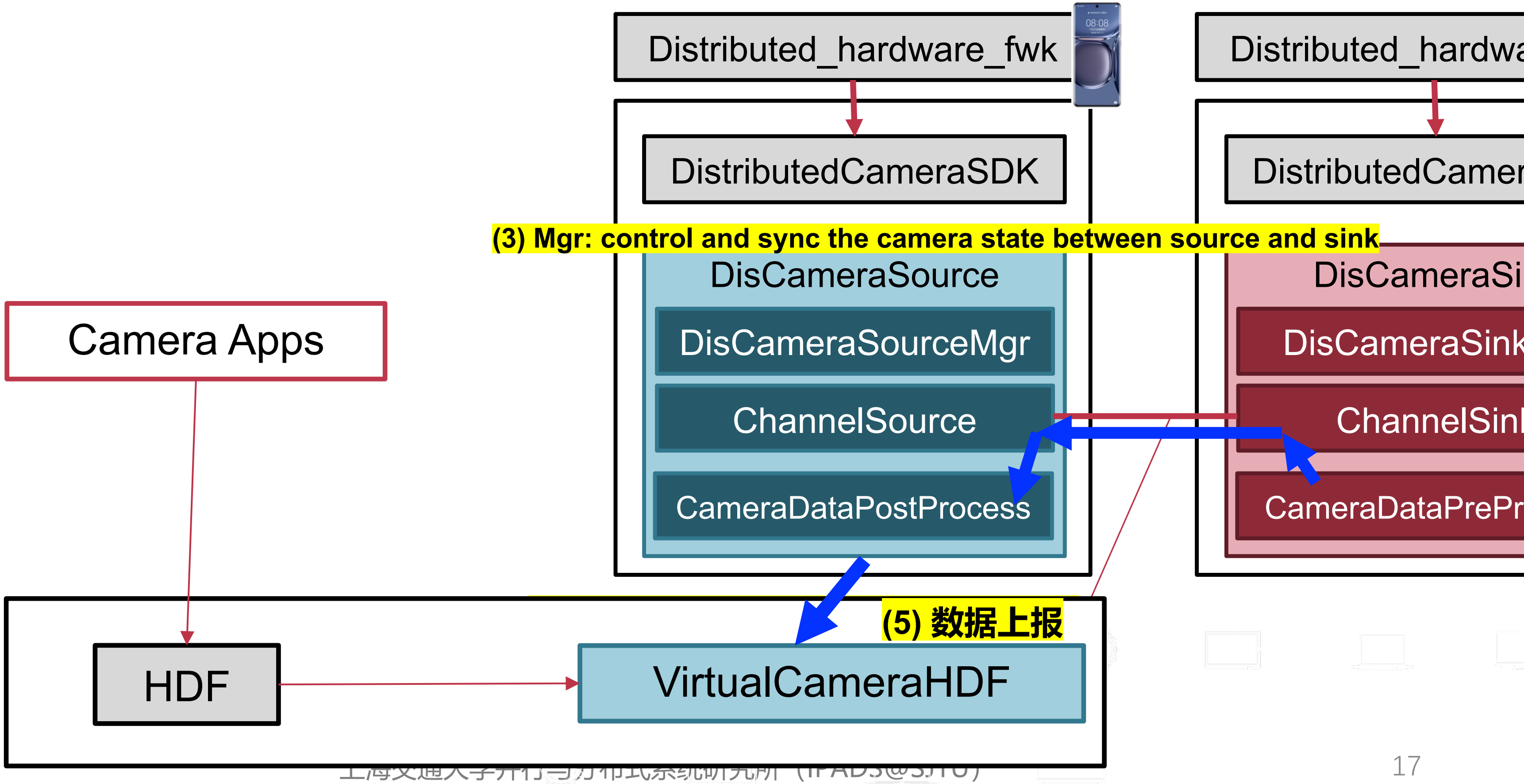
(1) Data collection and device operations

(2) Channel: software-bus based channel, exchange both data and metadata

(3) Mgr: control and sync the camera state between source and sink

(4) Camera data from sink to source

Distributed Hardware: Distributed Camera



(3) Mgr: control and sync the camera state between source and sink

(5) 数据上报

Distributed_hardware_fwk

DistributedCameraSDK

DisCameraSource

DisCameraSourceMgr

ChannelSource

CameraDataPostProcess

Distributed_hardwa

DistributedCamer

DisCameraSi

DisCameraSink

ChannelSinl

CameraDataPrePr

Camera Apps

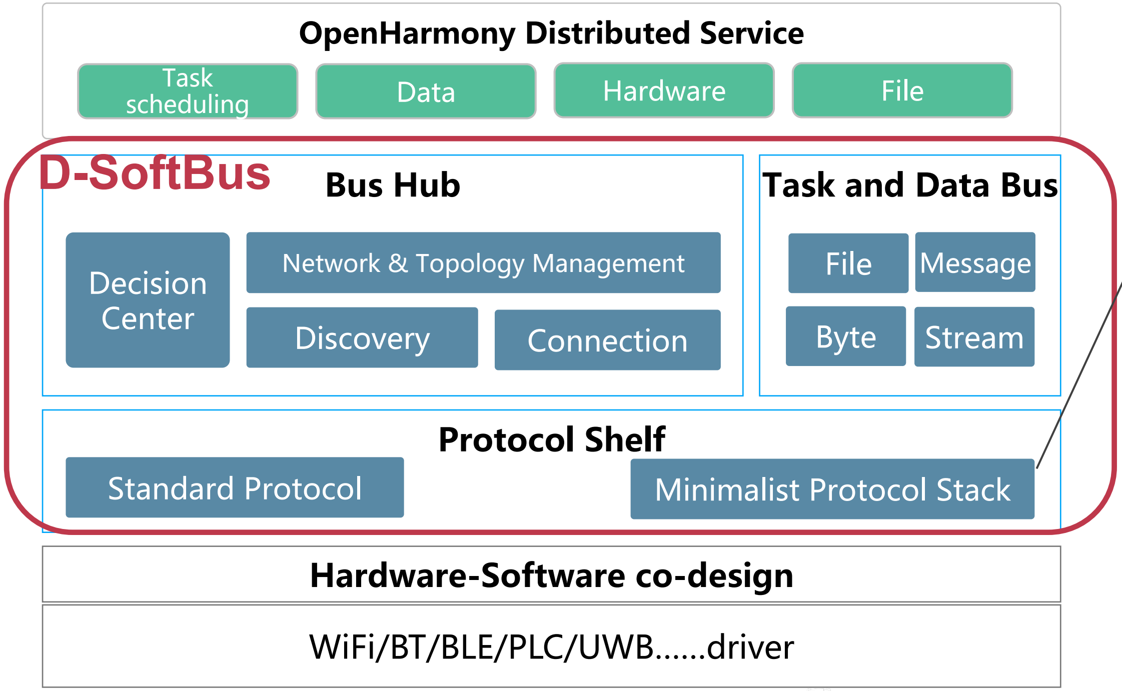
HDF

VirtualCameraHDF

DISTRIBUTED SOFT-BUS

Distributed Soft-Bus

Distributed Soft Bus for **discovery**, **connection**, and **data transfer**, providing a real-time online connectivity channel for mobiles/IoT



OpenHarmony Distributed Service

Task scheduling | Data | Hardware | File

D-SoftBus

Bus Hub

Decision Center | Network & Topology Management | Discovery | Connection

Task and Data Bus

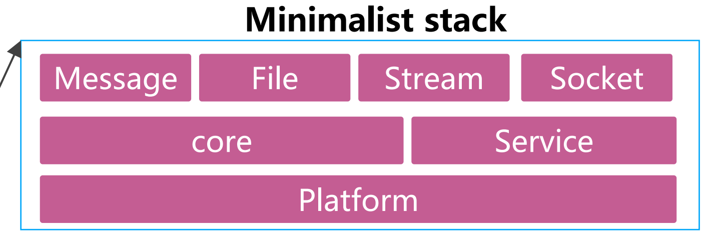
File | Message | Byte | Stream

Protocol Shelf

Standard Protocol | Minimalist Protocol Stack

Hardware-Software co-design

WiFi/BT/BLE/PLC/UWB……driver



Minimalist stack

Message | File | Stream | Socket

core | Service

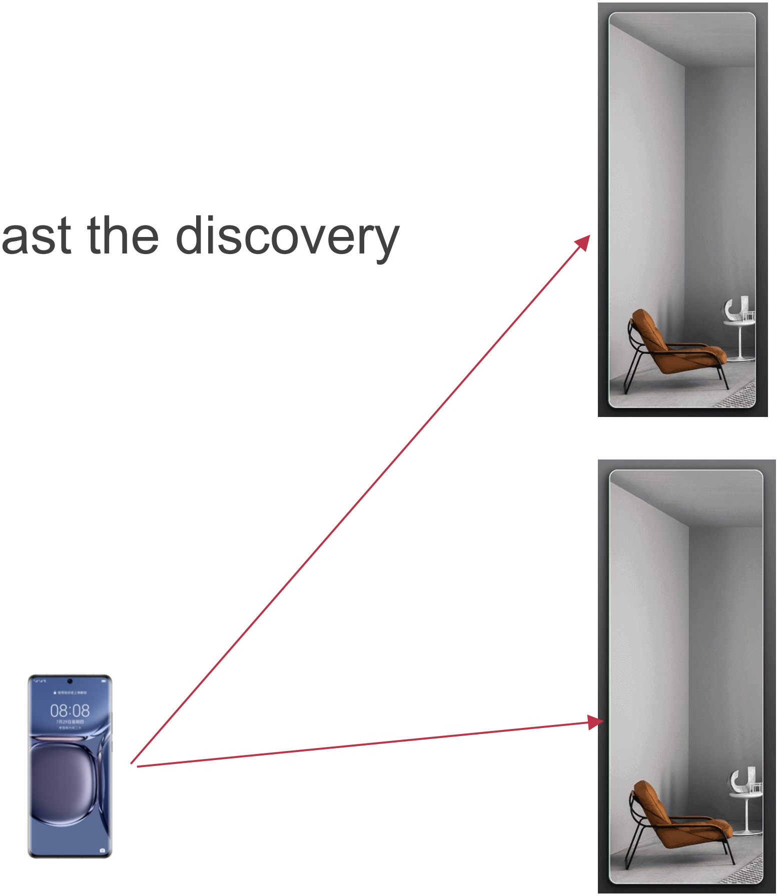
Platform

Eliminates **redundant header overhead**, leverages real-time MAC layer status to accurately sense **wireless network quality**, and adjusts packet transmission rate precisely

- **Compressed protocol layers**
- **Precision Congestion Control**
- **Chip Frequency Tuning/Energy Saving**

D-SoftBus: Device Discovery

- **Phone**
  - Using specific protocol to broadcast the discovery request



D-SoftBus: Discovery

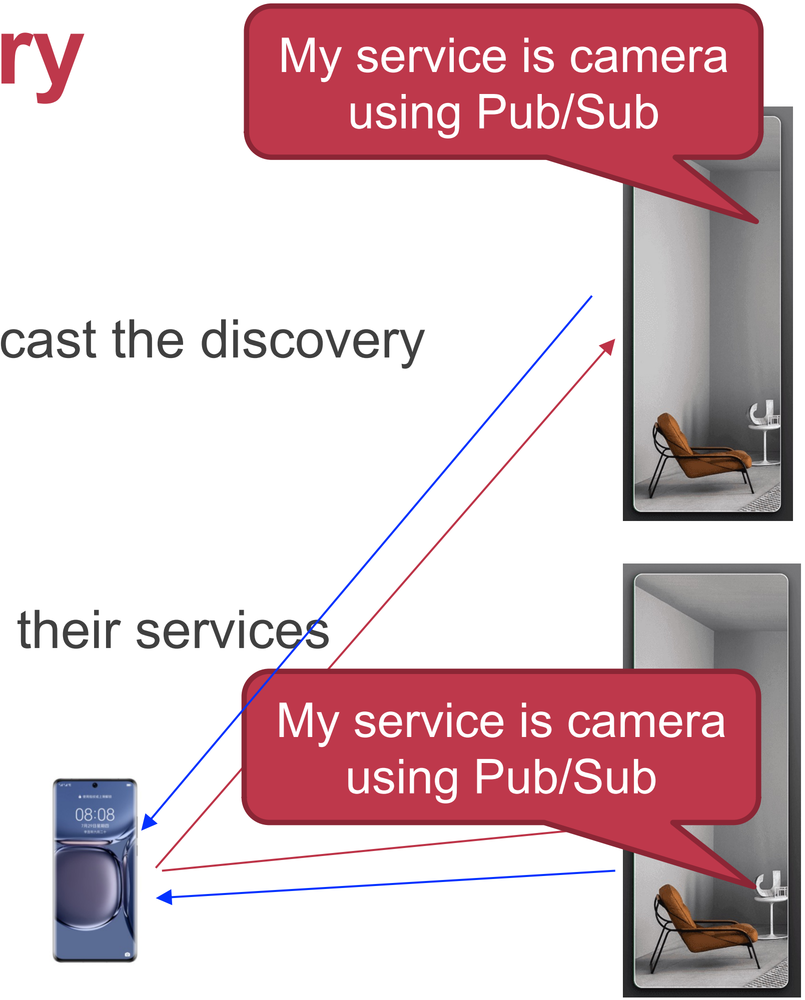
My service is camera using Pub/Sub

- **Phone**
  - Using specific protocol to broadcast the discovery request
- **Devices**
  - Using pub/sub model to expose their services

My service is camera using Pub/Sub

D-SoftBus: Discovery

- **Phone**
  - Using specific protocol to broadcast the discovery request
- **Devices**
  - Using pub/sub model to expose their services
  - Ack the requests



My service is camera using Pub/Sub

My service is camera using Pub/Sub

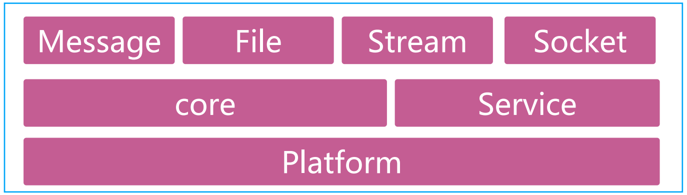
D-SoftBus: Discovery

- **Phone**
  - Using specific protocol to broadcast the discovery request
  - Update the D-SoftBus info
- **Devices**
  - Using pub/sub model to expose their services
  - Ack the requests

D-SoftBus: Transmission

- **Customized protocol**
  - Minimalist network stack
  - Other customizable (e.g., QUIC)
- **Multi-Path transport**
  - Utilize abilities of 4G/5G/WLAN/etc.
- **Intelligent latency control**
  - Optimize latency
- **Dynamic resource scheduling**

Minimalist stack



Message | File | Stream | Socket

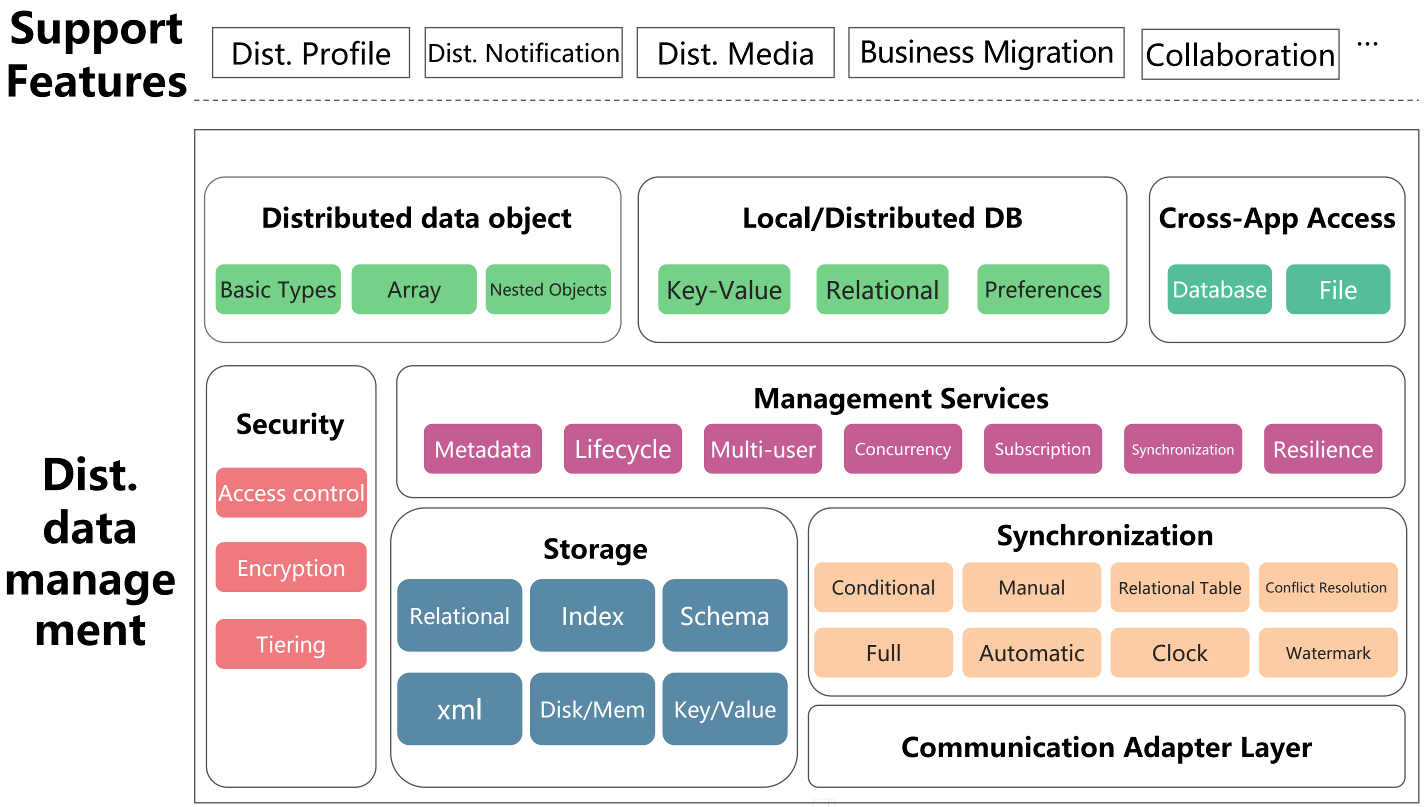
core | Service

Platform

Eliminates **redundant header overhead**, leverages real-time MAC layer status to accurately sense **wireless network quality**, and adjusts packet transmission rate precisely

- **Compressed protocol layers**
- **Precision Congestion Control**
- **Chip Frequency Tuning/Energy Saving**

Distributed Data



Support Features

Dist. Profile | Dist. Notification | Dist. Media | Business Migration | Collaboration …

Dist. data management

Distributed data object: Basic Types, Array, Nested Objects

Local/Distributed DB: Key-Value, Relational, Preferences

Cross-App Access: Database, File

Security: Access control, Encryption, Tiering

Management Services: Metadata, Lifecycle, Multi-user, Concurrency, Subscription, Synchronization, Resilience

Storage: Relational, Index, Schema, xml, Disk/Mem, Key/Value

Synchronization: Conditional, Manual, Relational Table, Conflict Resolution, Full, Automatic, Clock, Watermark

Communication Adapter Layer

Distributed object

Supported Types: JS Primitives/Arrays/Nested Objects

Distributed K-V

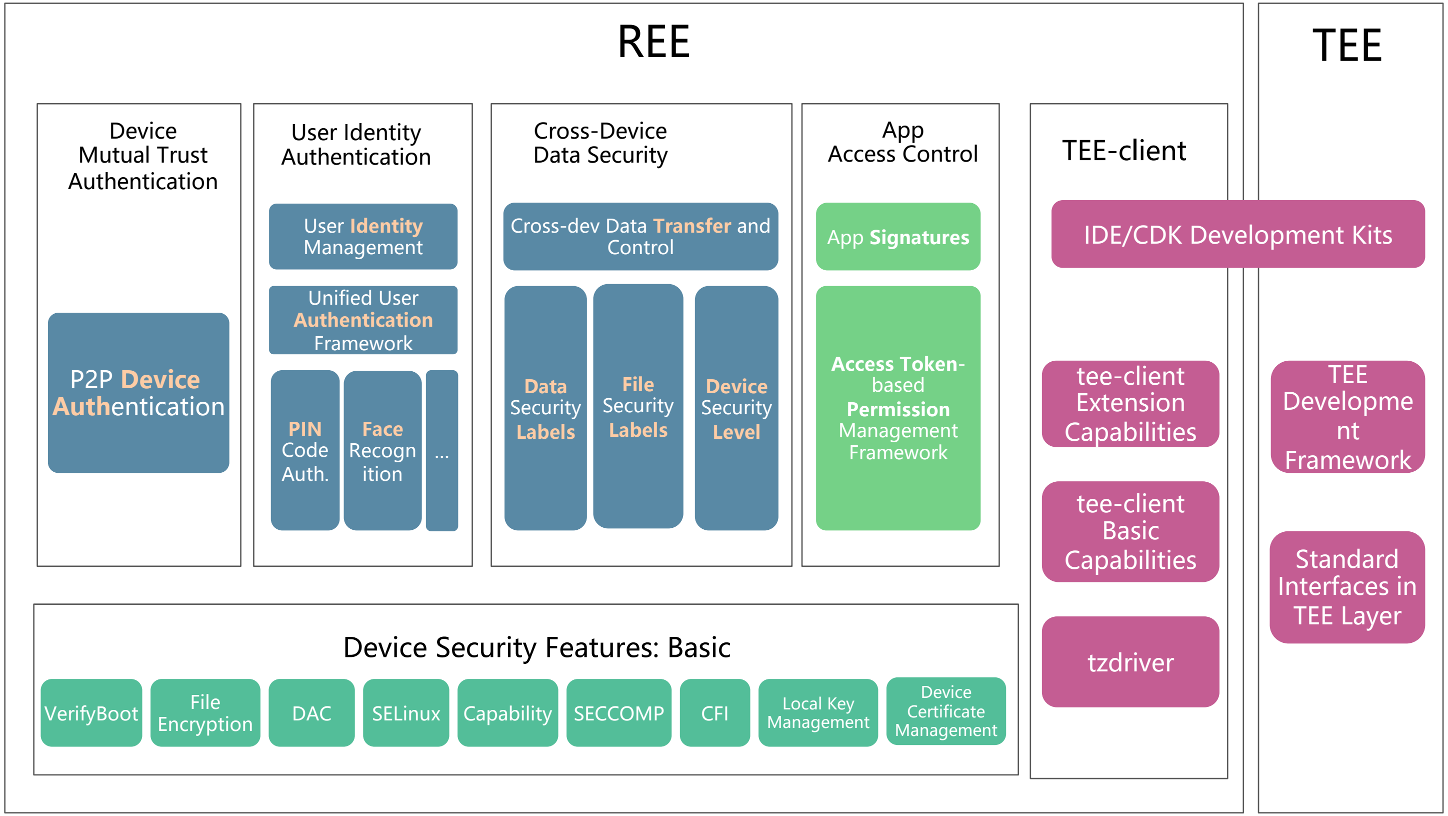
Synchronization Modes: Full Sync (Manual/Automatic)

Distributed relational

Synchronization Modes: Full & Conditional Sync (Manual)

Reliability & Security

Distributed Security Framework


REE
TEE
Device Mutual Trust Authentication
P2P Device Authentication
User Identity Authentication
User Identity Management
Unified User Authentication Framework
PIN Code Auth.
Face Recognition
...
Cross-Device Data Security
Cross-dev Data Transfer and Control
Data Security Labels
File Security Labels
Device Security Level
App Access Control
App Signatures
Access Token-based Permission Management Framework
TEE-client
IDE/CDK Development Kits
tee-client Extension Capabilities
tee-client Basic Capabilities
tzdriver
TEE Development Framework
Standard Interfaces in TEE Layer
Device Security Features: Basic
VerifyBoot
File Encryption
DAC
SELinux
Capability
SECCOMP
CFI
Local Key Management
Device Certificate Management

Distributed device mutual-trust authentication

Distributed user identity authentication

Cross-device data security

Access control

Trusted Execution Environment (TEE)

Summary

- **OpenHarmony takes distribution as the first-class capability**
  - Distributed hardware
  - Software bus
  - Distributed data and security
- **Try the exercise/demo for distribution in next talk ☺**