

06 April 2026



**CHERI**

# Securing Real-Time Communication

**Porting and running Edge Computing for  
mission critical services on Morello CHERI**

**Peter Cox**

CEO and Founder – UM Labs

**Alexander Voevoda**

Chief Engineer – UM Labs

# ○ Agenda

- ◆ About UM Labs
- ◆ Real-Time Communication Security
- ◆ Core/Edge Architecture
- ◆ Why Morello CHERI?
- ◆ Porting Target
- ◆ Porting Experiences
- ◆ Performance Comparison
- ◆ Conclusion and Future Work

## ○ About UM Labs

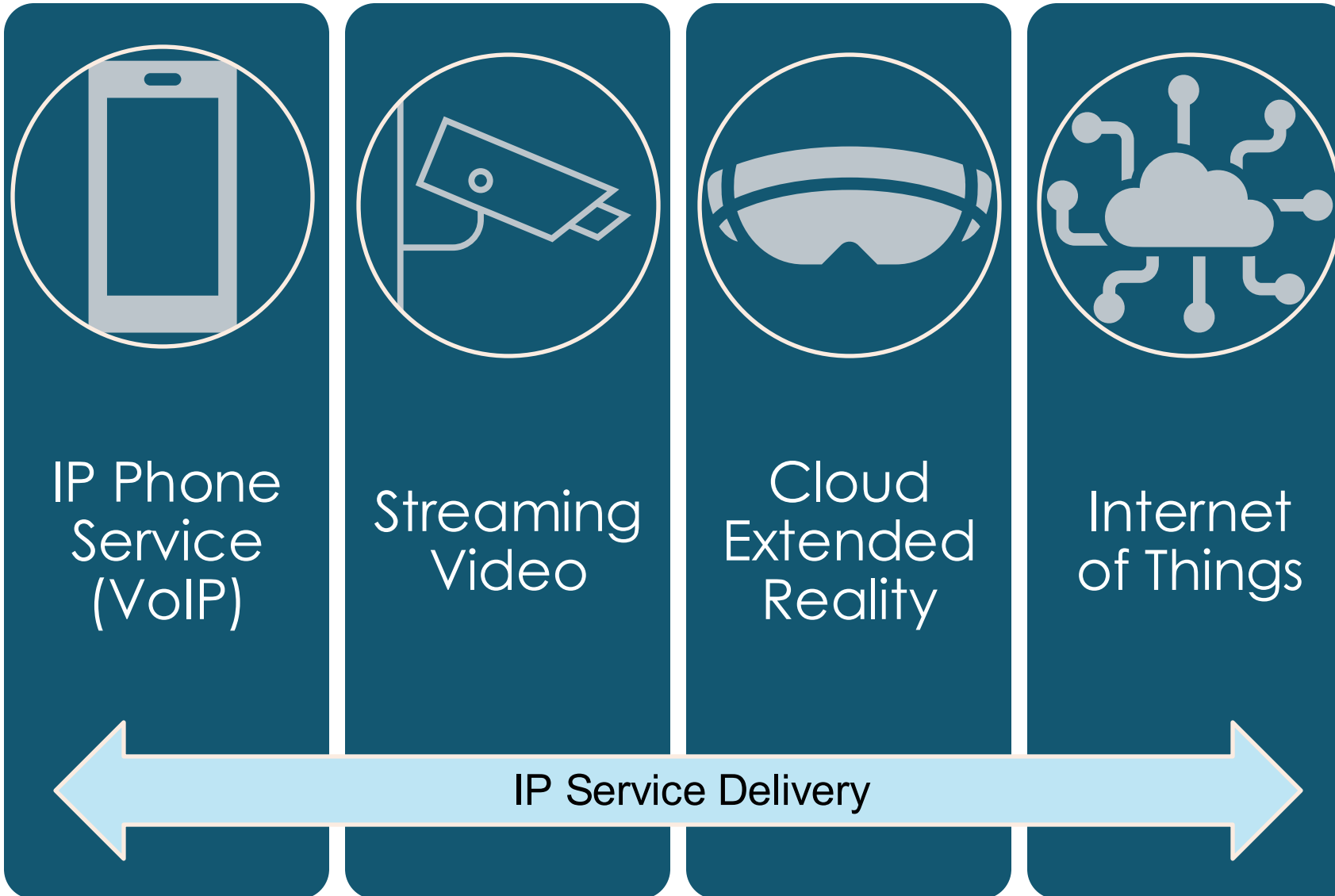


- UK based Cybersecurity R&D for Real-Time Communication on IP Networks
- Technology operational in multiple government, defence and commercial networks
- Working with major European Telcos and systems integrators

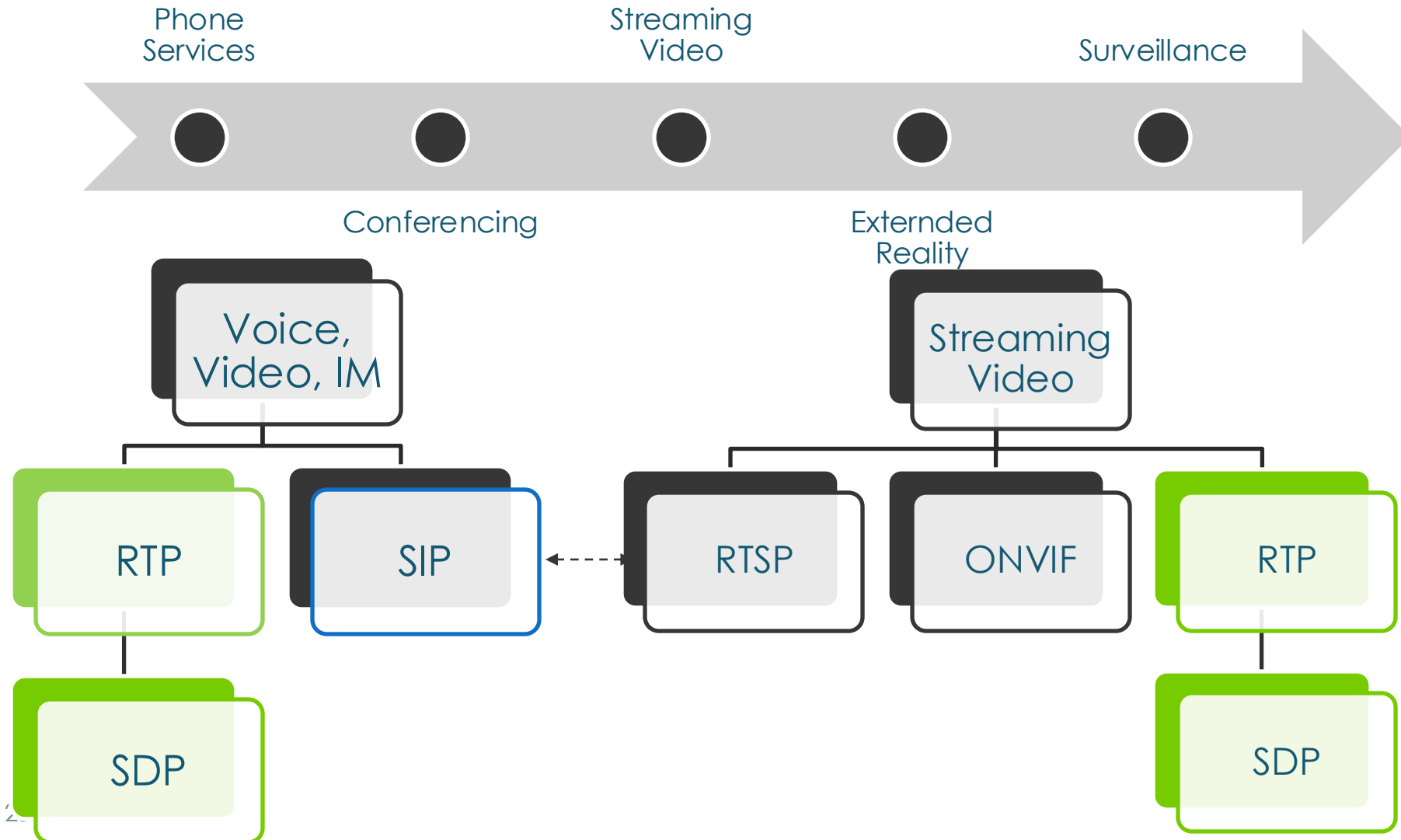
<https://www.um-labs.com>

This presentation makes references to concepts and designs which are the intellectual property of UM Labs

# ○ Real-Time Communication (RTC) on IP Networks



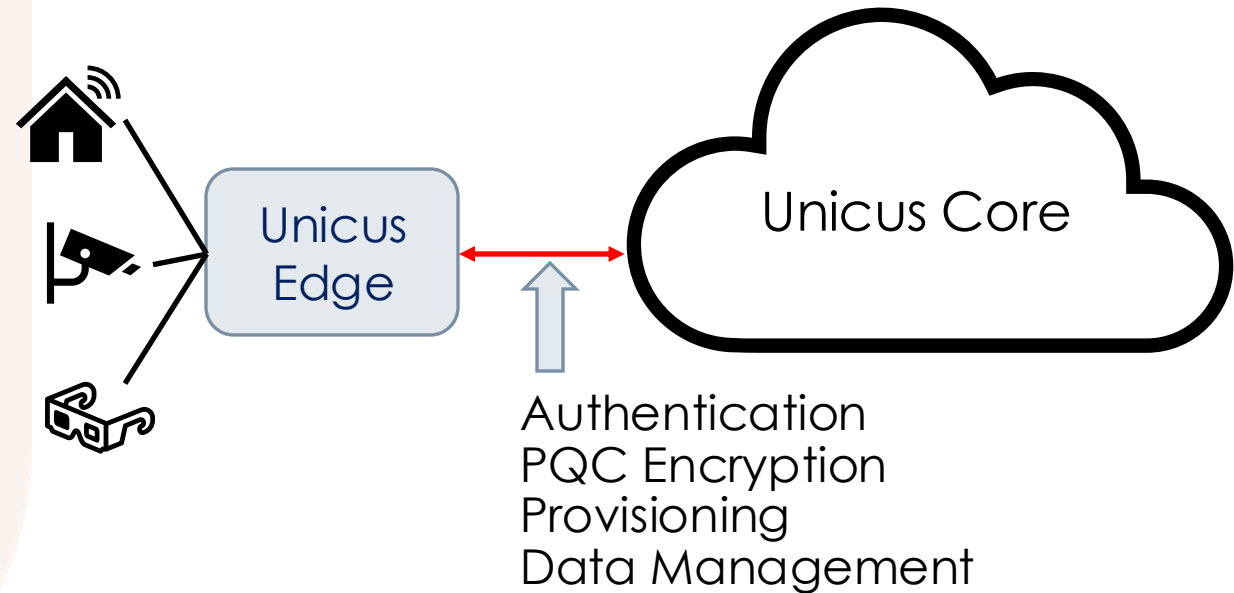
# RTC: Multiple Protocols



06 April 2024

# Unicus® -- Core/Edge Architecture

- Core in any public/private cloud
- Security limited on many end-devices
- Edge protects those devices and secures connection to Core
- Edge -- small scale system, Linux, BSD Unix, iOS, Android
- Core/Edge common shared, layered architecture
- Full stack security



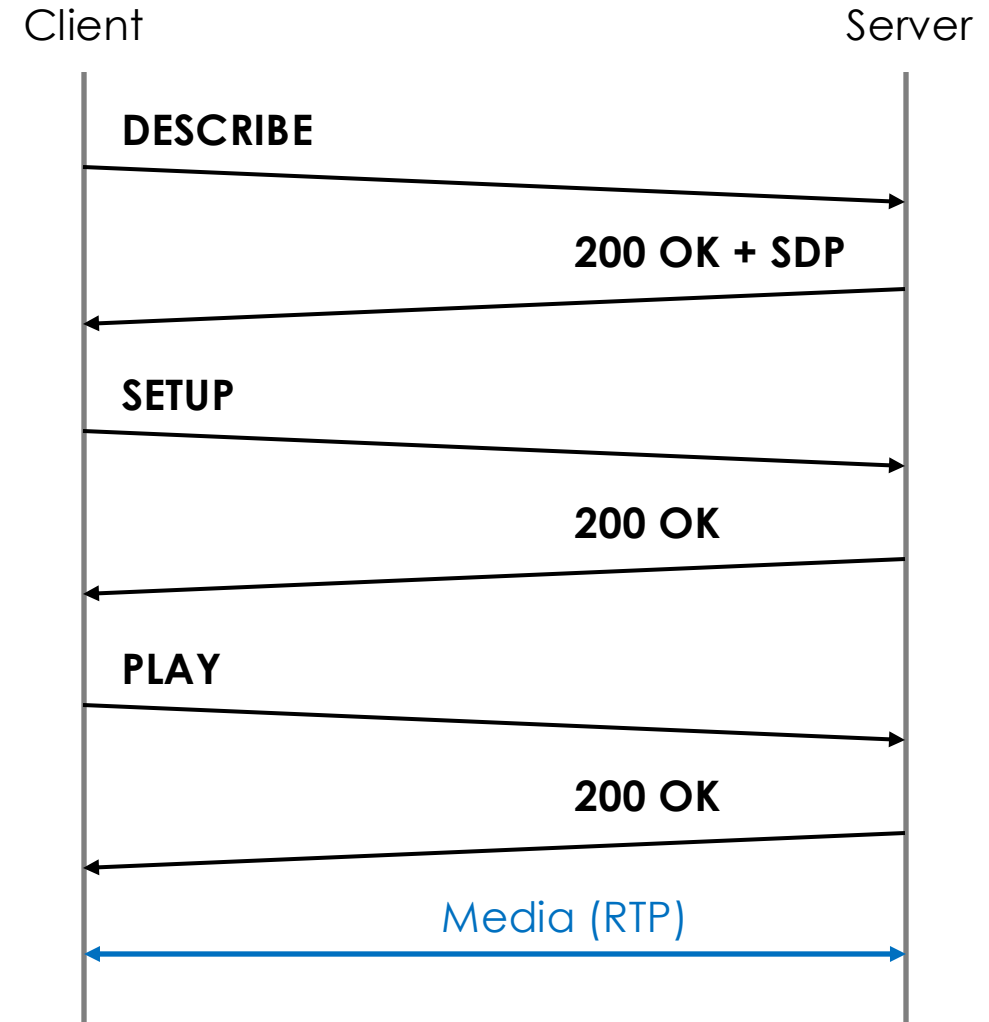


## ○ Why Morello CHERI?

- Robust software design and secure implementation essential for Cybersecurity products
- Port to CheriBSD provides
  - Additional assurance on the security of our design and implementation
  - Ability to assess CHERI as a suitable deployment platform for applications with elevated security requirements

# Porting Target

- Unicus Edge running SVSS
- Steaming Video Security Service
- RTSP (RFC 2326), session setup
- SDP (RFC 4566) media stream negotiation
- RTP (RFC 3550) media transfer





# ○ Porting Experience

## Development + Validation

- ◆ Morello QEMU via cheribuild
- ◆ SLIRP blocks all RFC 1918 addresses -> SSH tunnel mandatory
- ◆ End-to-end verified, PQC, RTSP streaming, loss = 0%
- ◆ QEMU tuned: SMP=4, 1 GB RAM, TB cache 1 GB → 0% packet loss

## Silicon Validation

- ◆ Arm Morello SDP board
- ◆ Package deployment, core dump analysis, rc.d daemon management
- ◆ CPU/memory metrics via FreeBSD sysctl APIs

Timescale: **2 months** · **~330** engineer-hours



## ○ Build Environment

- Cross-compilation: macOS M1 + Ubuntu host → CheriBSD/Morello target
- `CHERIBSD_DEVEL`: `-O0`, `-Wcheri-pedantic`, `-cheri-bounds=very-aggressive`
- Pre-porting analysis (native Linux):
  - ASAN, UBSAN, TSAN, scan-build, cppcheck, valgrind
- During porting (CheriBSD):
  - `-Wcheri`, `-Wcheri-capability-misuse`, `-Wcheri-prototypes`



# ○ Software Changes Required

## Unicus Edge

- ~67 CHERI related commits

Category	Commits
QEMU/Tunnel	~8
Morello Hardware	~12
3 <sup>rd</sup> Party Deps	~12
Unicus Edge Code	~35

Net new CHERI code < 5%  
code base

## 3<sup>rd</sup> Party Packages

- 7 Ported, no CHERI pkg available
  - GLib 2.76.4
  - GLib-Networking 2.80.1
  - JSON-GLib 1.10.8
  - LibFFI (purecap ABI)
  - LibSoup (GLIB\_VERSION patch)
  - guri.c (CHERI-safe URI parser)
  - OpenSSL 3.5.5 (PQC, no-asm)



# ○ Software Faults Detected

Unicus Edge, 9 Faults

Pointer arithmetic / sub-object underflow	3
Null termination(buffer too small)	1
Unaligned memory access (-Wcast-align)	4
Platform API incompatibility	1

3<sup>rd</sup> Party pkgs, 7 patched

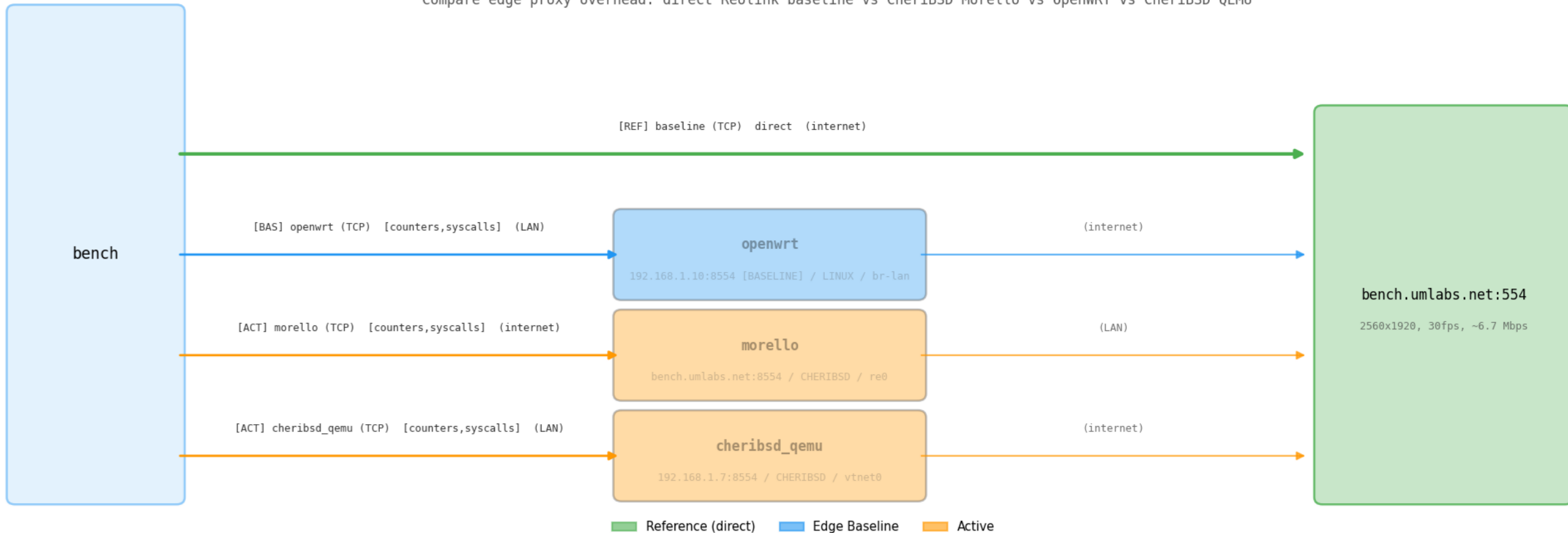
CHERI capability size (128 vs 64 bit)	3	Glib, JSON-Glib, Glib-Networking
Purecap ABI incompatibility	2	LibFFI Go closures, LibSoup GType
Capability Fault in URI parsing	1	guri.c replaces g_uri_split()
System OpenSSL lacks PQC support	1	cross-compiled 3.5.5



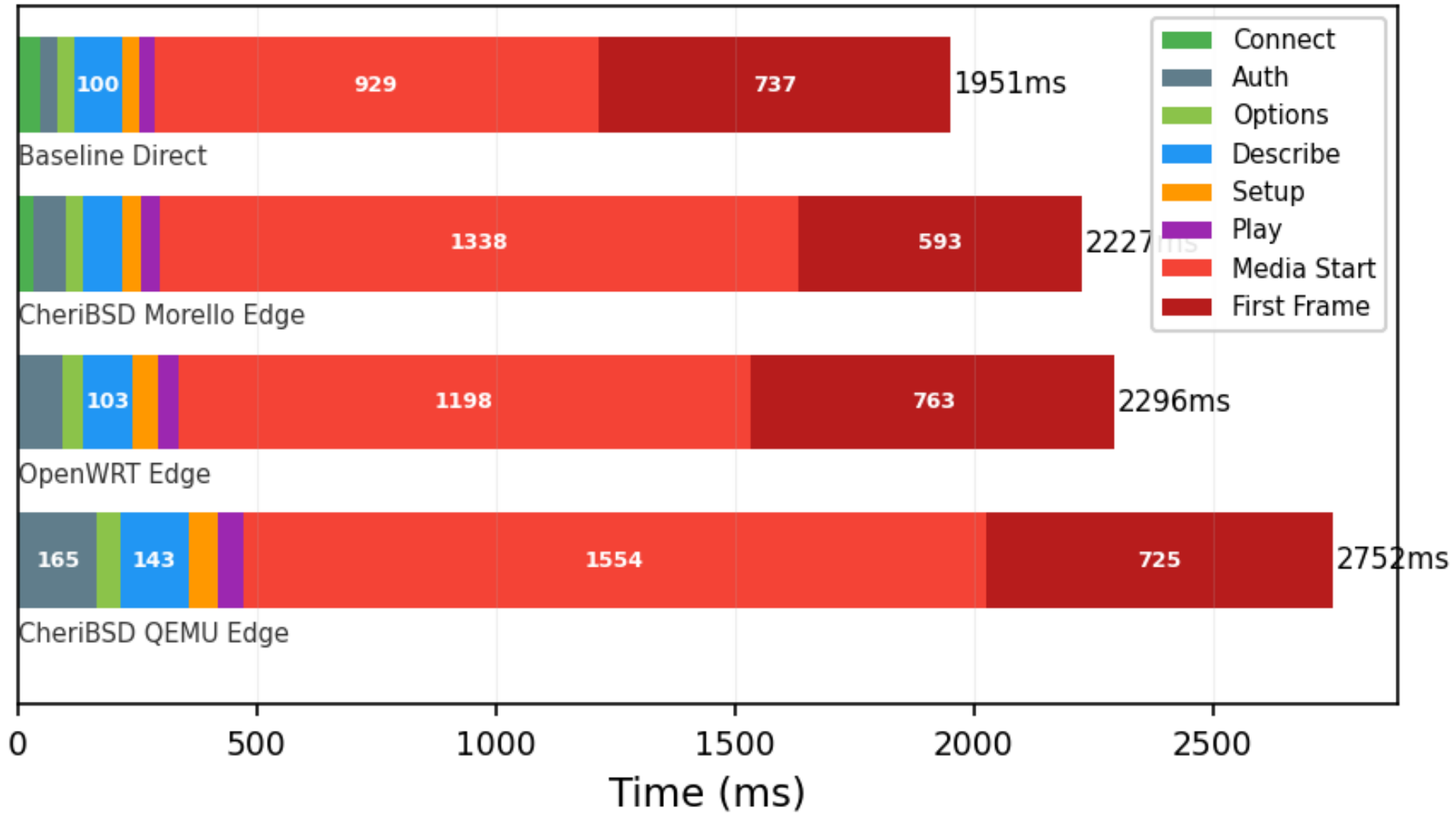
# Benchmark Topology

## Edge Platform Comparison - Morello / OpenWRT / CheriBSD QEMU

Compare edge proxy overhead: direct Reolink baseline vs CheriBSD Morello vs OpenWRT vs CheriBSD QEMU



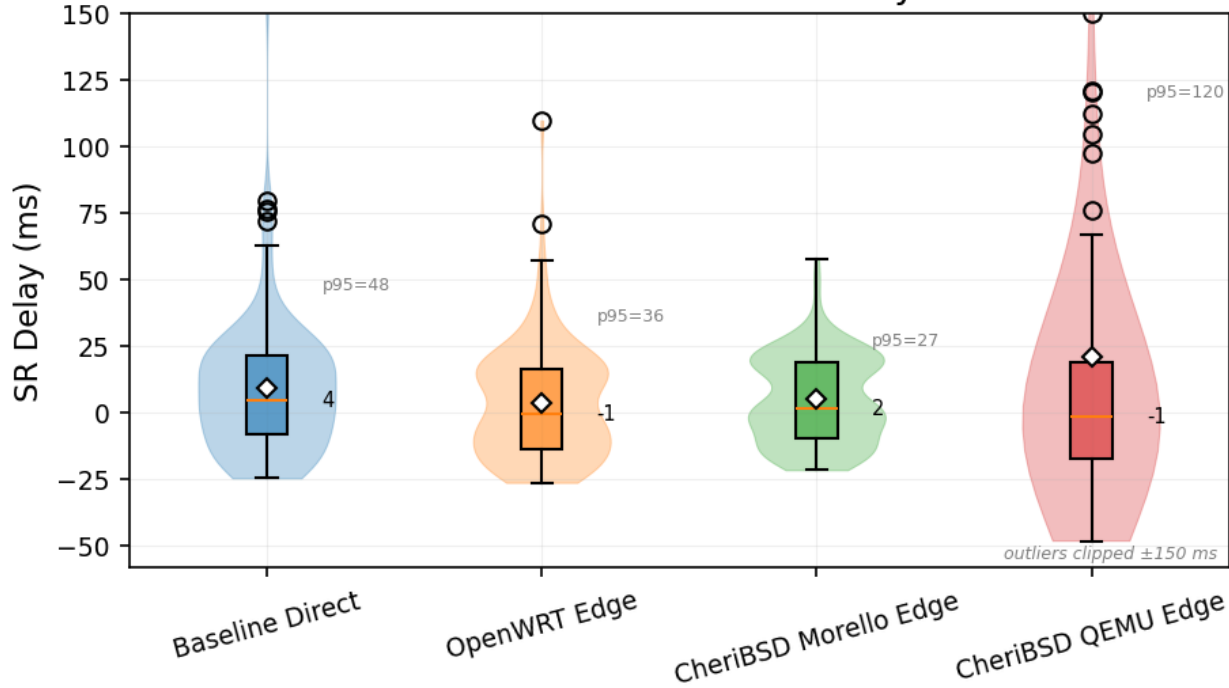
# RTSP Session Setup (Connect → First Frame)



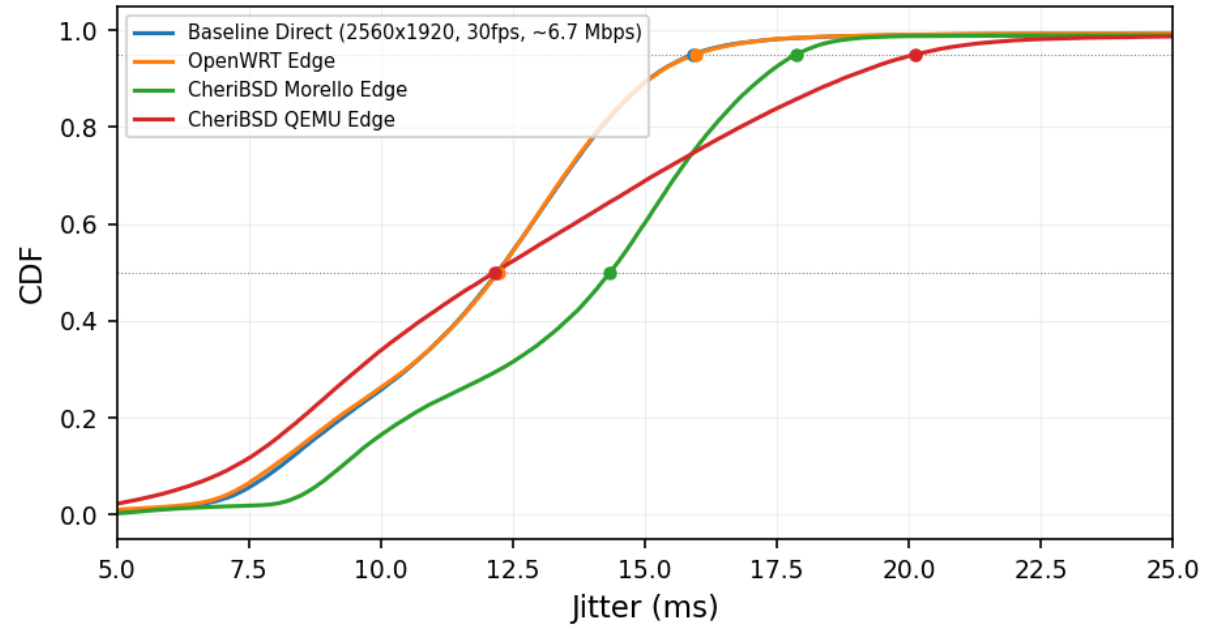
# Delay and Jitter



### Camera-to-Client Latency



### RTP Jitter CDF (RFC 3550)





# ○ Conclusion and Future Work

## Goals Met

- ◆ Morello port identified potential coding problems
  - ◆ UM Labs Code
  - ◆ 3<sup>rd</sup> Party Packages
- ◆ Morello port suitable for deployment
  - ◆ Performance comparable to OpenWRT Router
  - ◆ Elevated security requirements
  - ◆ Subject to cost and form factor

## Future Work

- ◆ Continued testing
- ◆ Further ports for code validation

# ○ Demo and Questions

