

09 April 2025



CHERIoT

2025 Update

Author

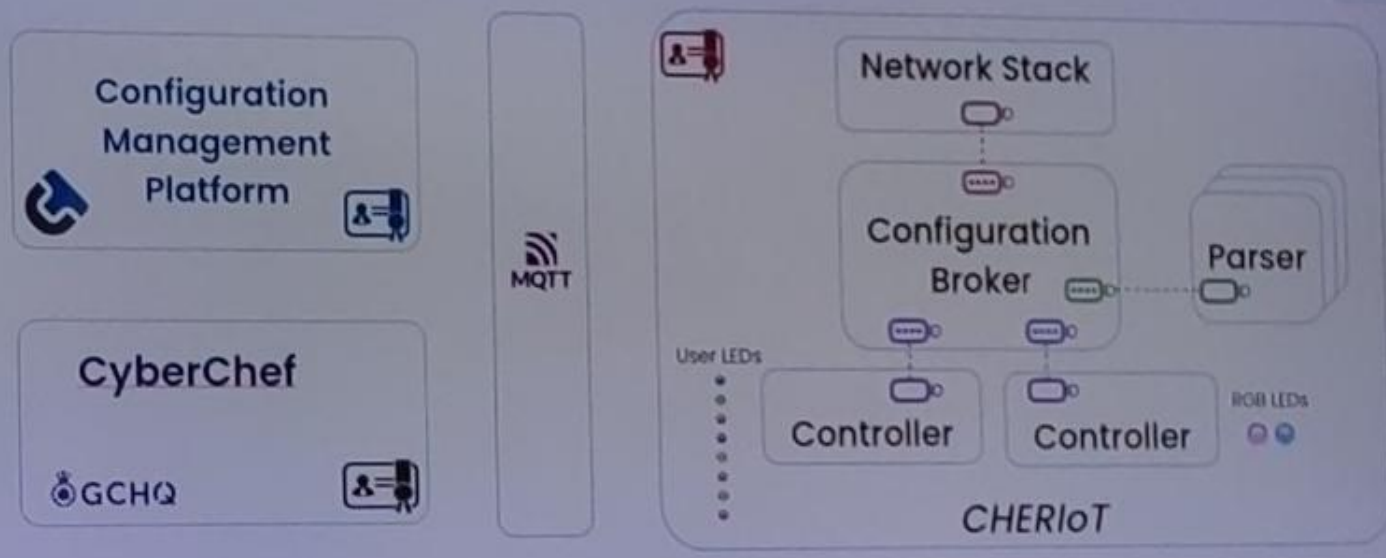
David Chisnall – SCI Semiconductor



○ **CHERIoT recap**

- ◆ First publicly announced at CHERITech '22
- ◆ Scales CHERI down to *tiny* devices
- ◆ Rich compartmentalisation model
- ◆ Showcases what you can do if you can assume CHERI
- ◆ Open source project: ISA specification, RTOS, toolchain, auditing tools, reference implementation of the core
- ◆ Maintainers: David Chisnall (SCI Semiconductor), Ben Laurie (Google), Yucong Tao (Microsoft)

Solution Overview



© Configured Things Ltd.

5

Protecting random numbers against memory attacks

Even with truly quantum, cryptographically secure random numbers, if an attacker **can access them in the memory**, the randomness is **compromised**.

That's why we combine **quantum entropy** with **CHERIoT's compartmentalization model**—to ensure keys stay **unpredictable and unreachable**.

Quantum Dice © 2025 restricted and proprietary 27/03/2025

April 2025





○ **CHERIoT ISA v1.0 and beyond**

- ◆ ISA frozen for 1.0 release
- ◆ Some typesetting issues to resolve before 'official' release
- ◆ Current draft available:
- ◆ <https://cheriot.org/cheriot-sail/cheriot-architecture.pdf>
- ◆ The 1.0 release will have no functional changes from this version

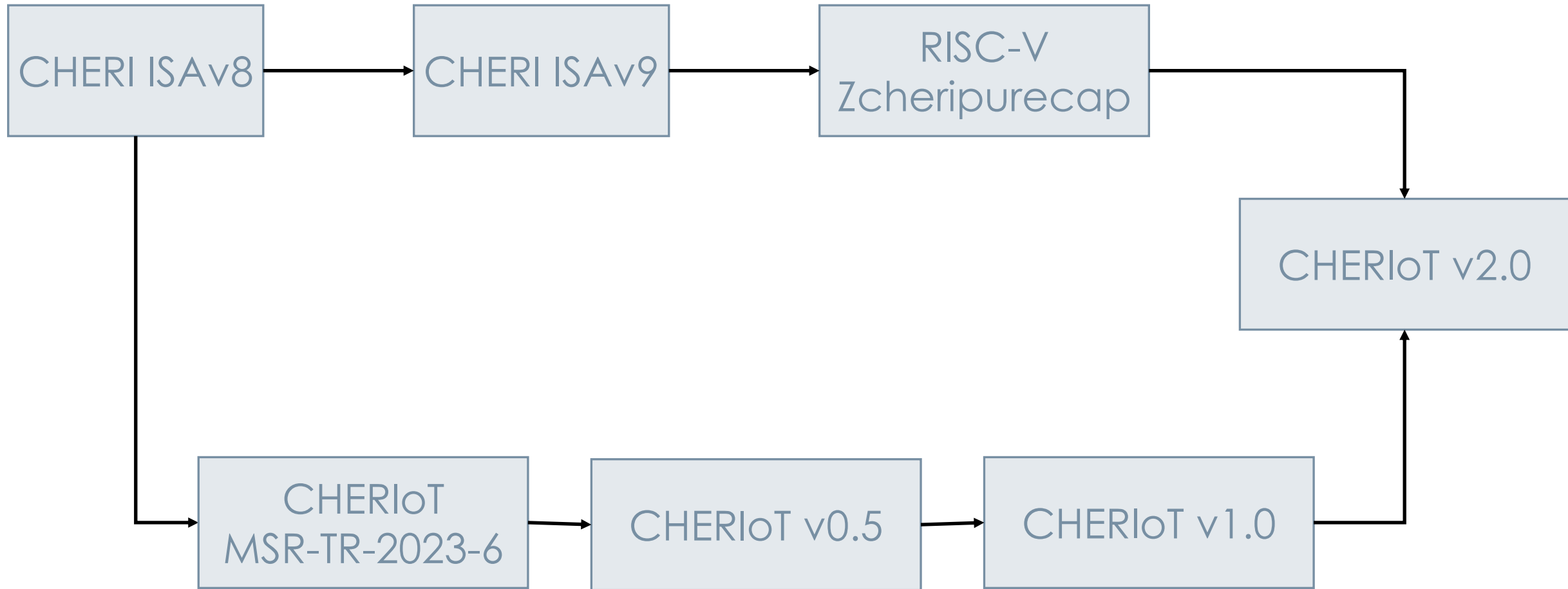


○ Return sentries in CHERIoT 1.0

- CHERIoT uses sentries for interrupt control
- Structured-programming for interrupt-disabled code
- Retrofitted return sentries without needing new instructions
- CFI within a compartment

Target	Link	Used for	Valid sentries
\$cra	\$cnull	Function return	Backward sentries
Not \$cra	\$cnull	Tail call	Unsealed or interrupt-inheriting forward sentries
Any	Not \$cnull or \$cra	Outlined code	“
Any	\$cra	Function call	Unsealed or forward sentries

○ The road to CHERIoT v2





○ **Goals for CHERIoT v2.0**

- ◆ Rebased on Zcheripurecap
- ◆ 100% source compatible with CHERIoT v1.0
- ◆ Assembly compatible via macro shims
- ◆ Encodings aligned with Zcheripurecap (Not binary compatible)



○ Improved error handling

```
CHERIOT_DURING
```

```
// Memory-safety error
```

```
a[x] = 12;
```

```
CHERIOT_HANDLER
```

```
// Code here runs on exception
```

```
CHERIOT_END_HANDLER
```

```
// Execution resumes here
```

○ Sealing is now part of the C/C++ type system!



```
int operation(struct SomeType *__sealed_capability x)
{
    if (struct SomeType *unsealed = token_unseal(key, x)
    {
        // Do something with unsealed
        return 0;
    }
    return -EINVAL;
}
```



○ CHERIoT-Audit: audit firmware

```
$ cheriot-audit --board sonata.json \  
  --firmware-report 03.https_example.json \  
  --module network_stack.rego \  
  -q 'data.network_stack.all_connection_capabilities'  
  
[  
  { "capability": { "connection_type": "UDP", "host":  
    "pool.ntp.org", "port": 123 }, "owner": "SNTP" },  
  { "capability": { "connection_type": "TCP", "host":  
    "example.com", "port": 443 }, "owner": "https_example" }  
]
```

○ **CHERIoT Handbook First Edition**

- ◆ 236 pages (excluding contents / index)
- ◆ Completed technical review and copyediting
- ◆ Print editions available for purchase soon
- ◆ Live drafts available online: <https://cheriot.org/book/>



○ SCI ICENI chips coming soon!



○ **More questions? Join our public chat on Signal**

