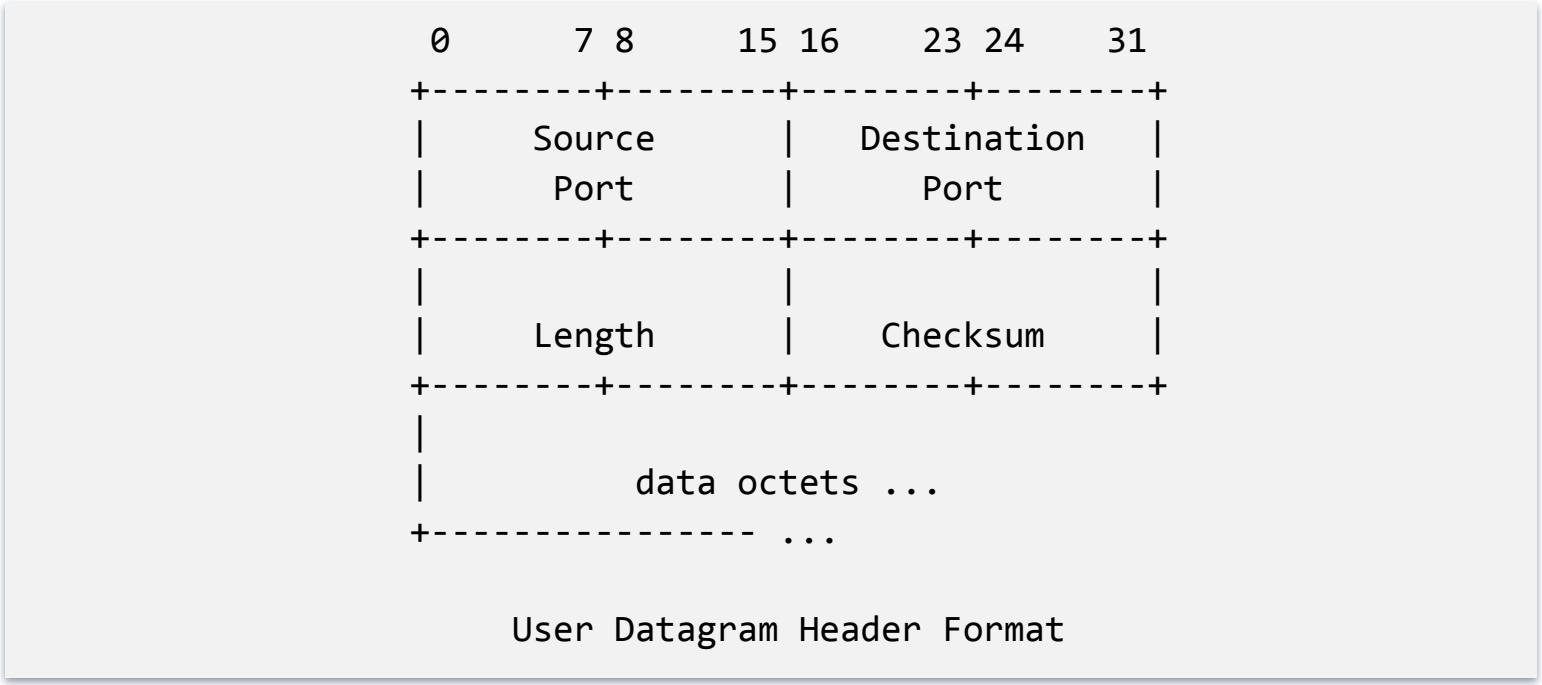
Hardening NVIDIA's Confidential Computing

A Formally Verified Implementation of the SPDM Device Attestation Protocol

Tobias Reiher

Mar 28, 2024

A fairly simple protocol: UDP



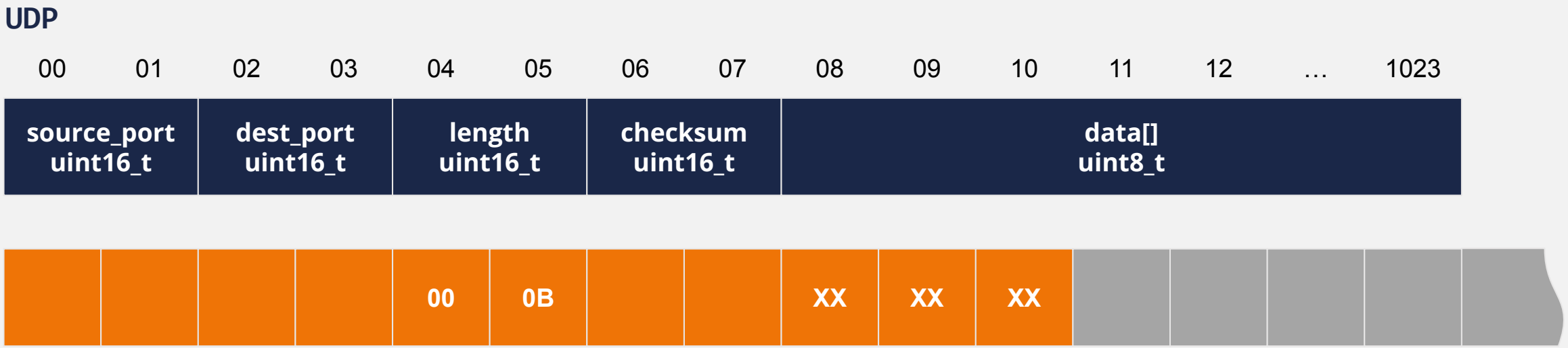
User Datagram Header Format

"Length is the length in octets of this user datagram including this header and the data." RFC 768

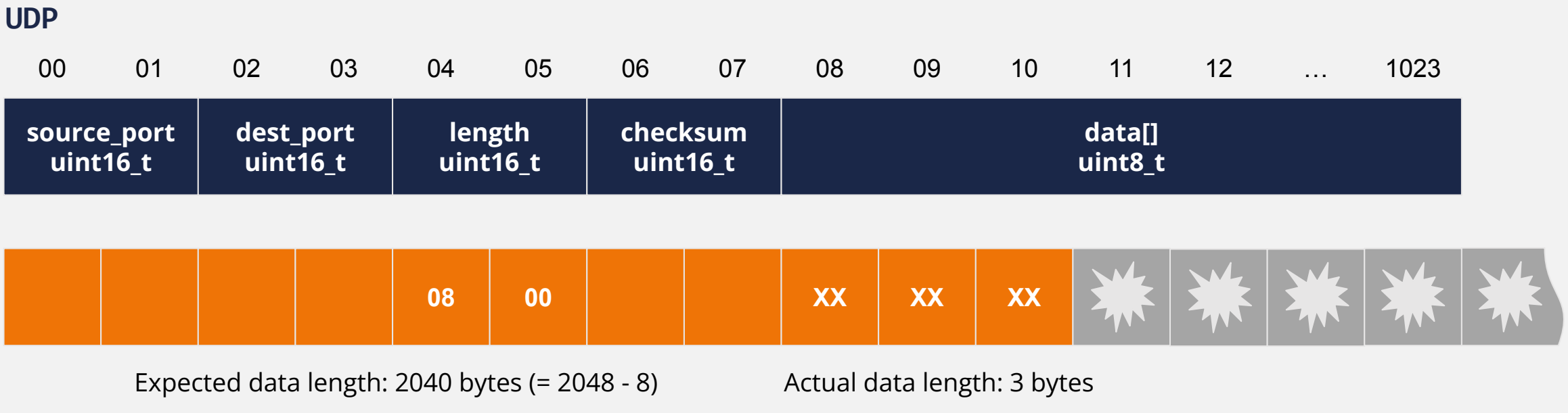
Typical representation with a C struct

UDP

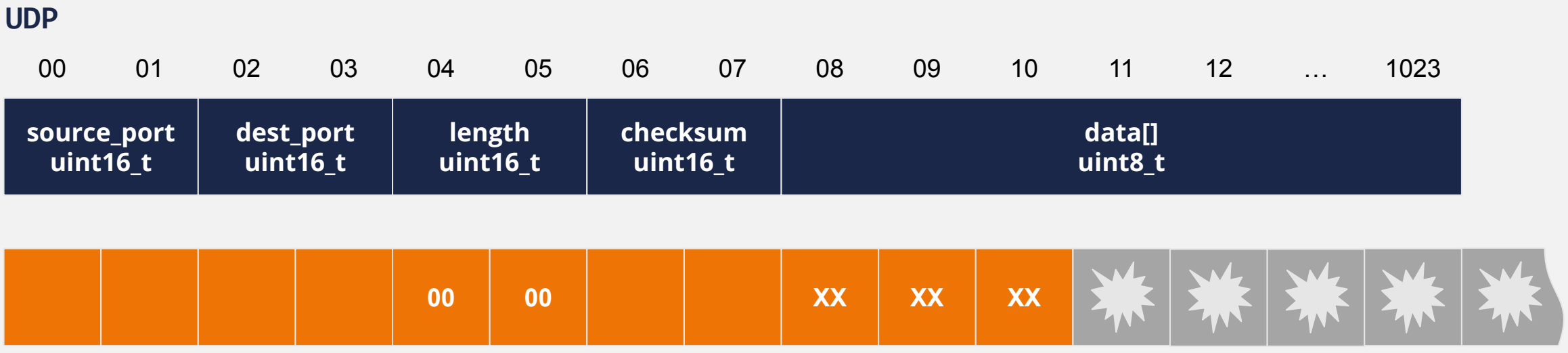
Typical representation with a C struct



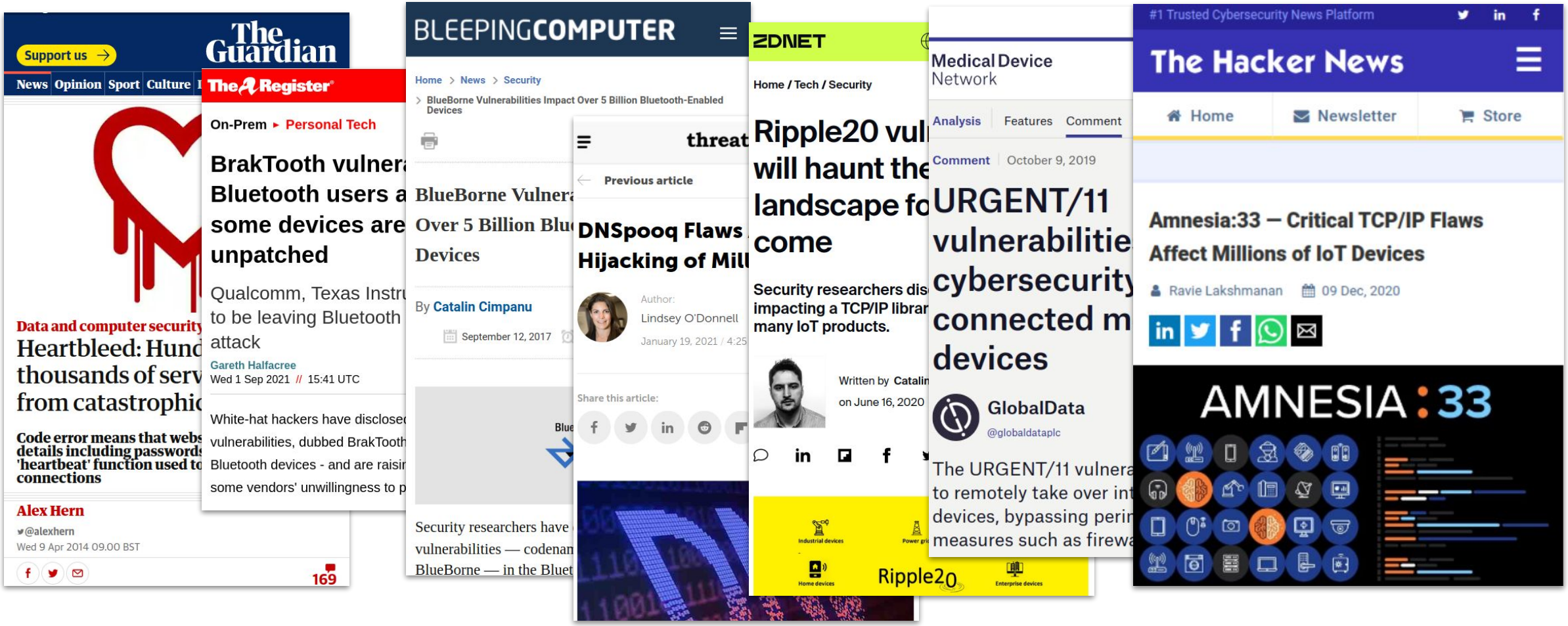
What could go wrong?



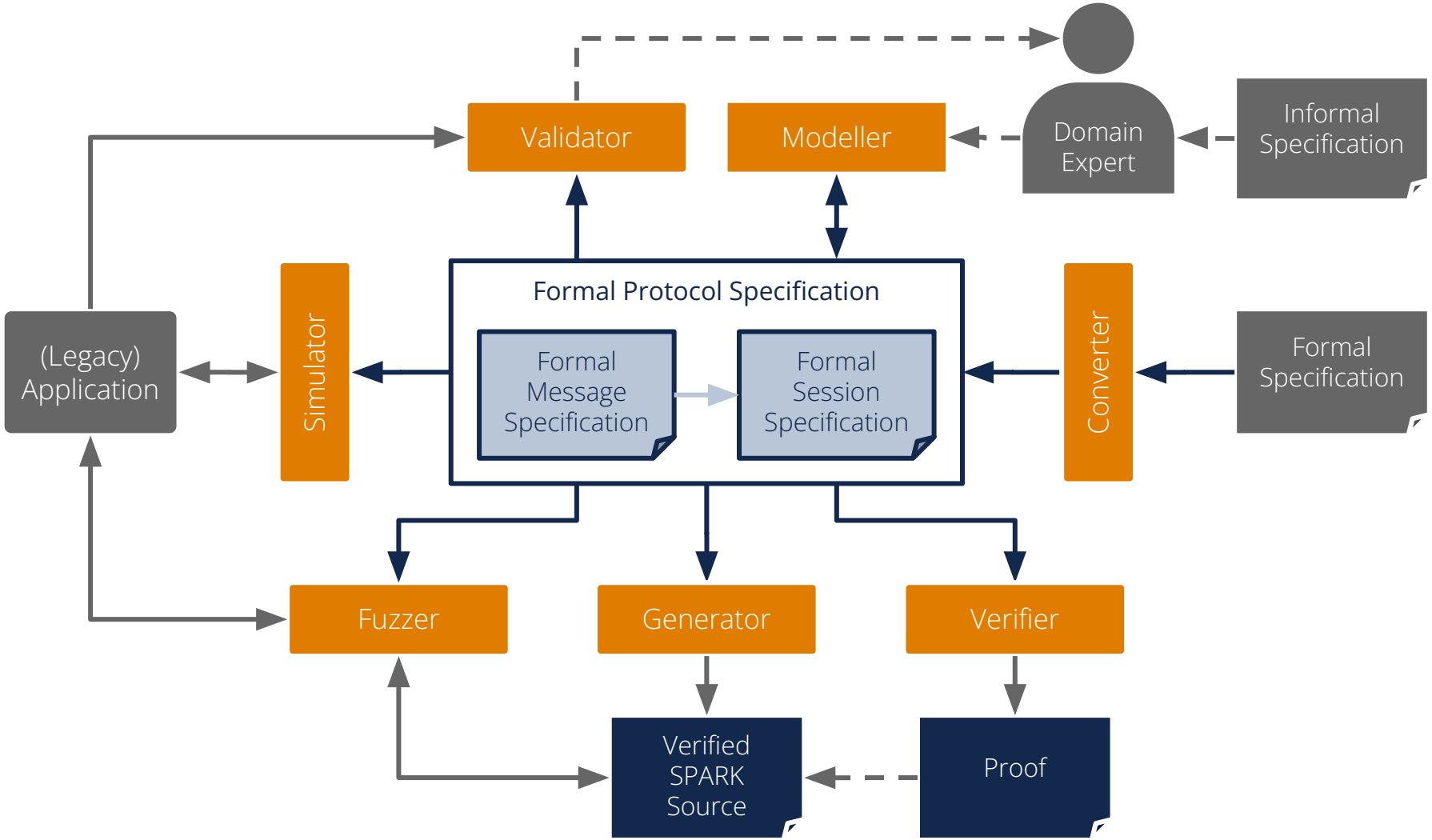
What could go wrong?



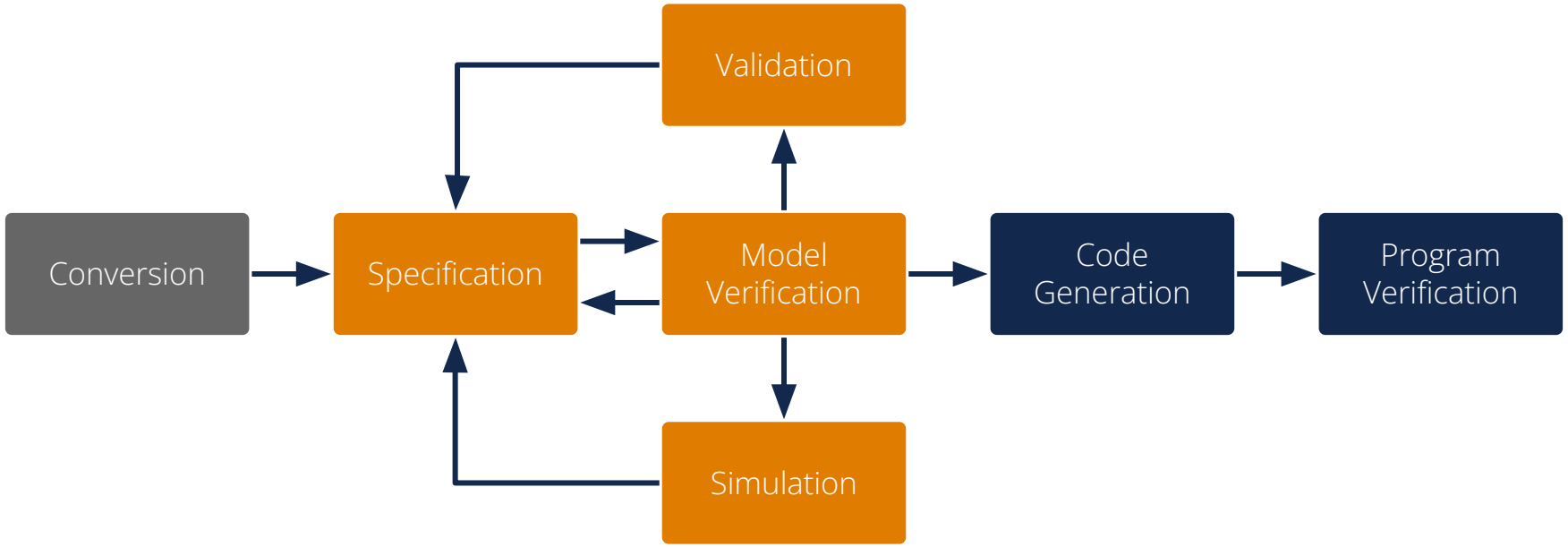
Real consequences and huge costs

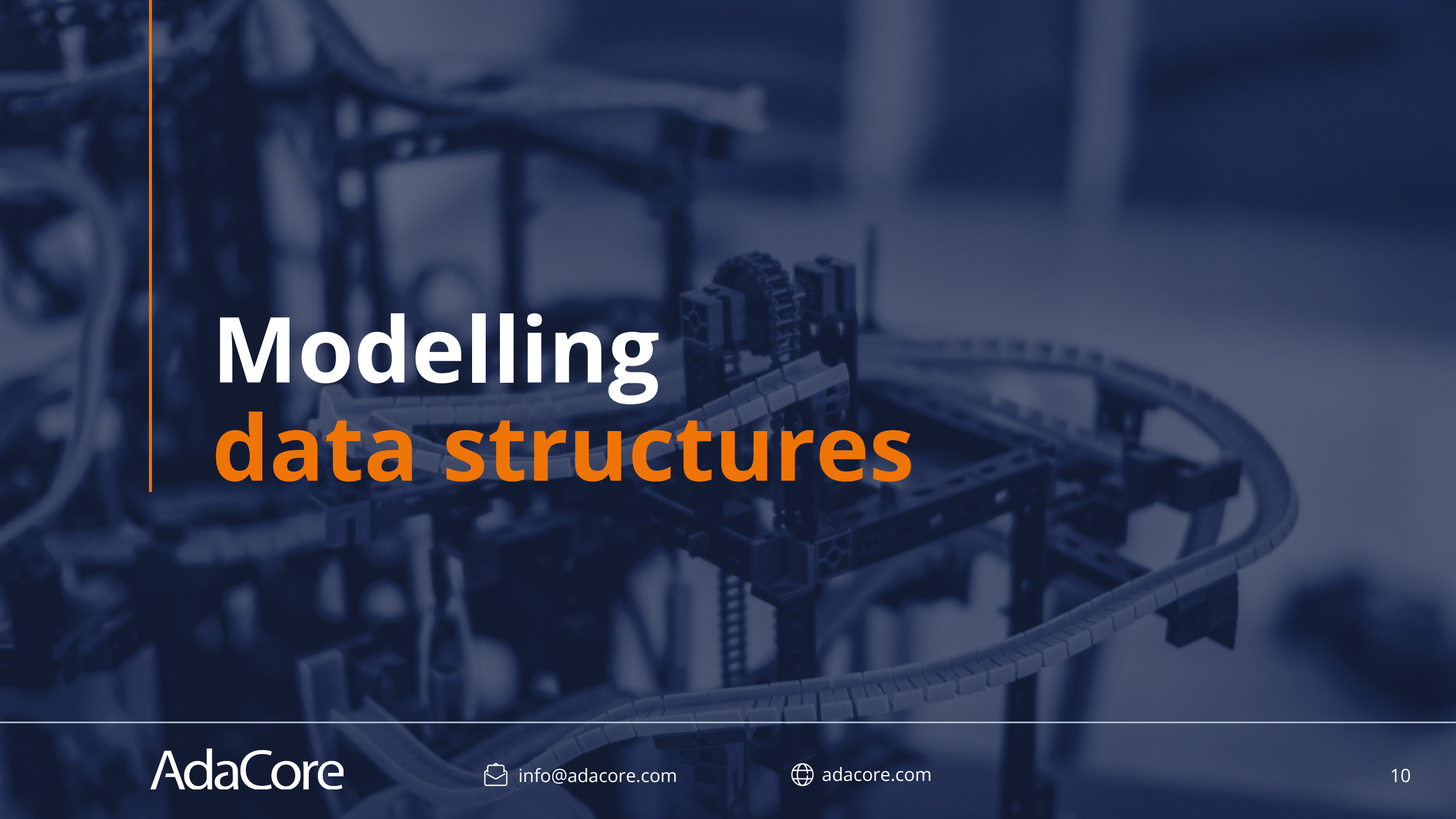


RecordFlux toolset



RecordFlux workflow

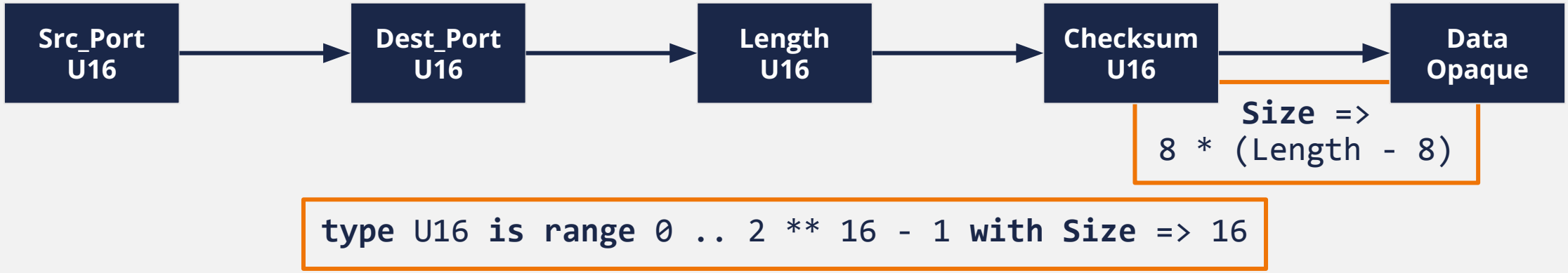




Modelling
data structures

Formal message specification

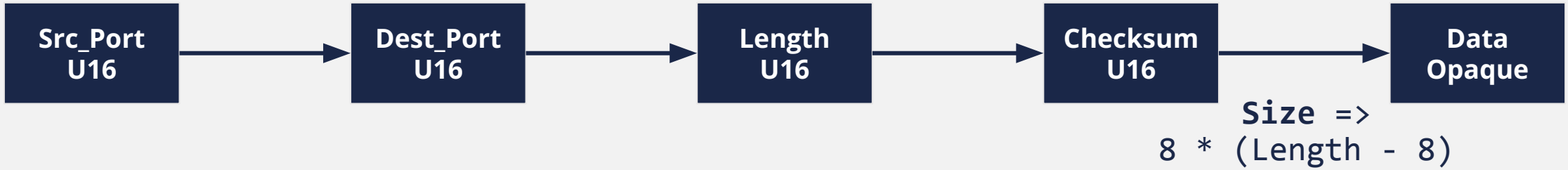
UDP




```
type U16 is range 0 .. 2 ** 16 - 1 with Size => 16
```

Proofs on the model level

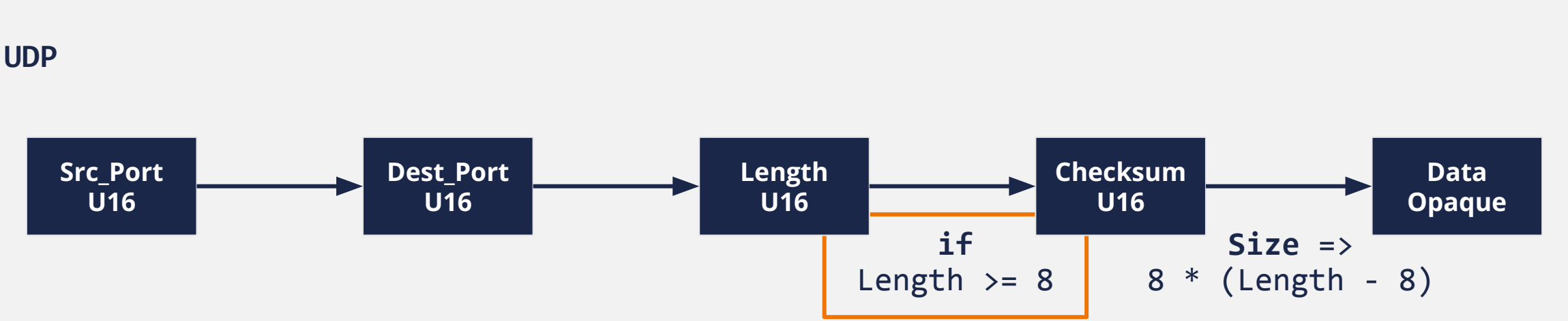
UDP




```
type U16 is range 0 .. 2 ** 16 - 1 with Size => 16
```

```
model: error: negative size for field "Data"
(Src_Port -> Dest_Port -> Length -> Checksum -> Data)
```

Option #1: Introduce explicit invariant



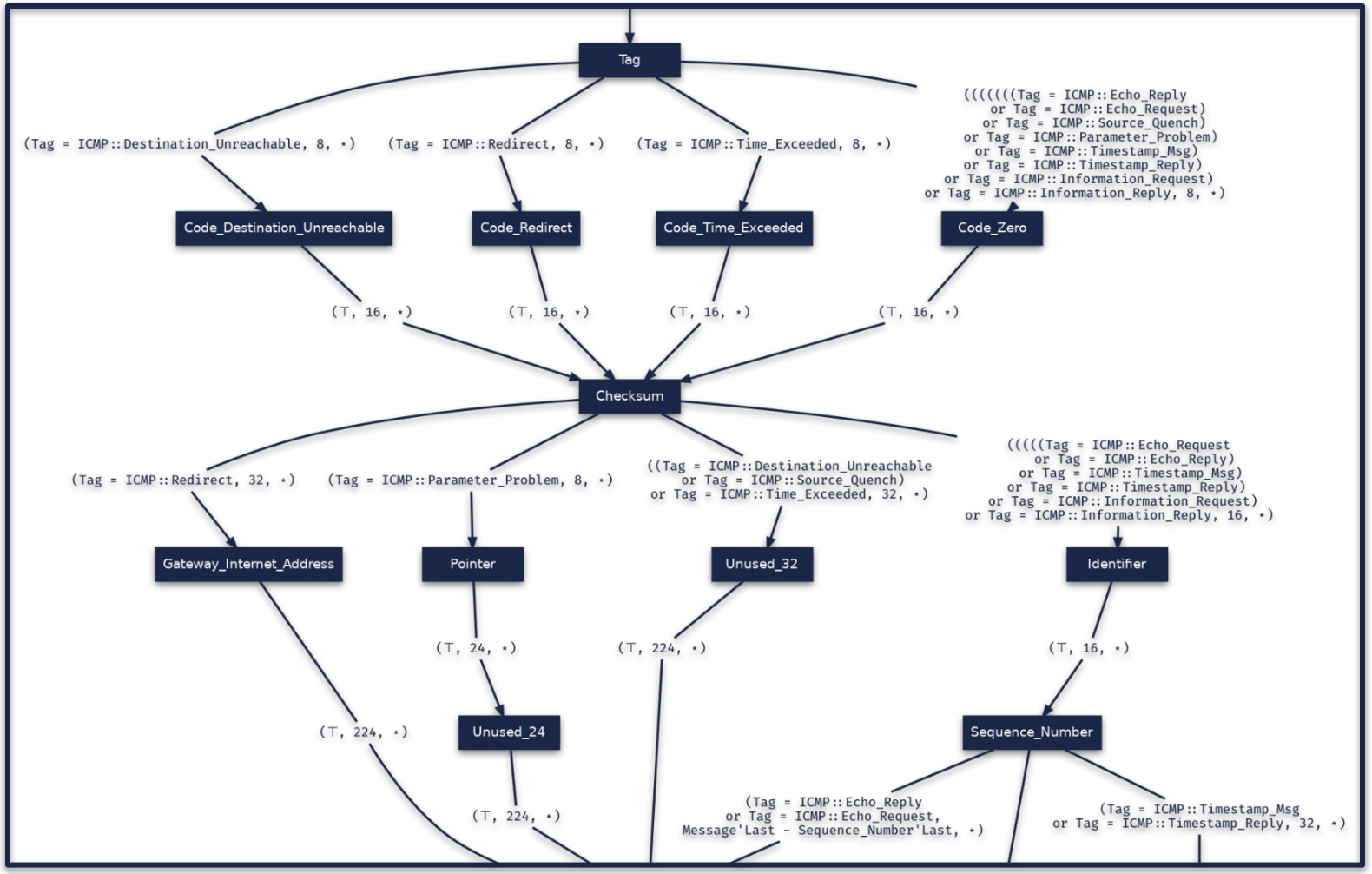
Option #2: Use constraint length type

UDP


```
type U16 is range 0 .. 2 ** 16 - 1 with Size => 16
```

```
type Length_T is range 8 .. 2 ** 16 - 1 with Size => 16
```

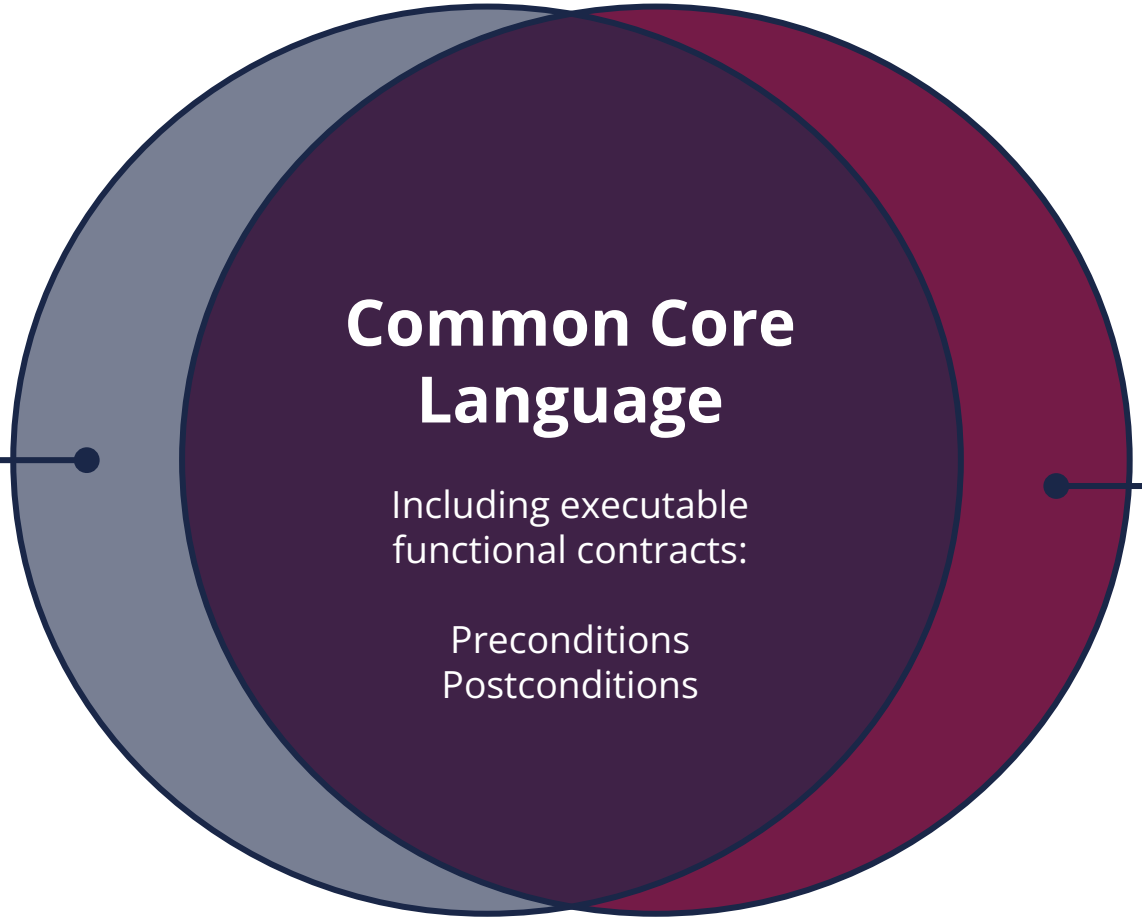
A fragment of an ICMP message model



Guarantees from message verification

How do we bridge the gap between specification and implementation?

SPARK as our target language



Ada

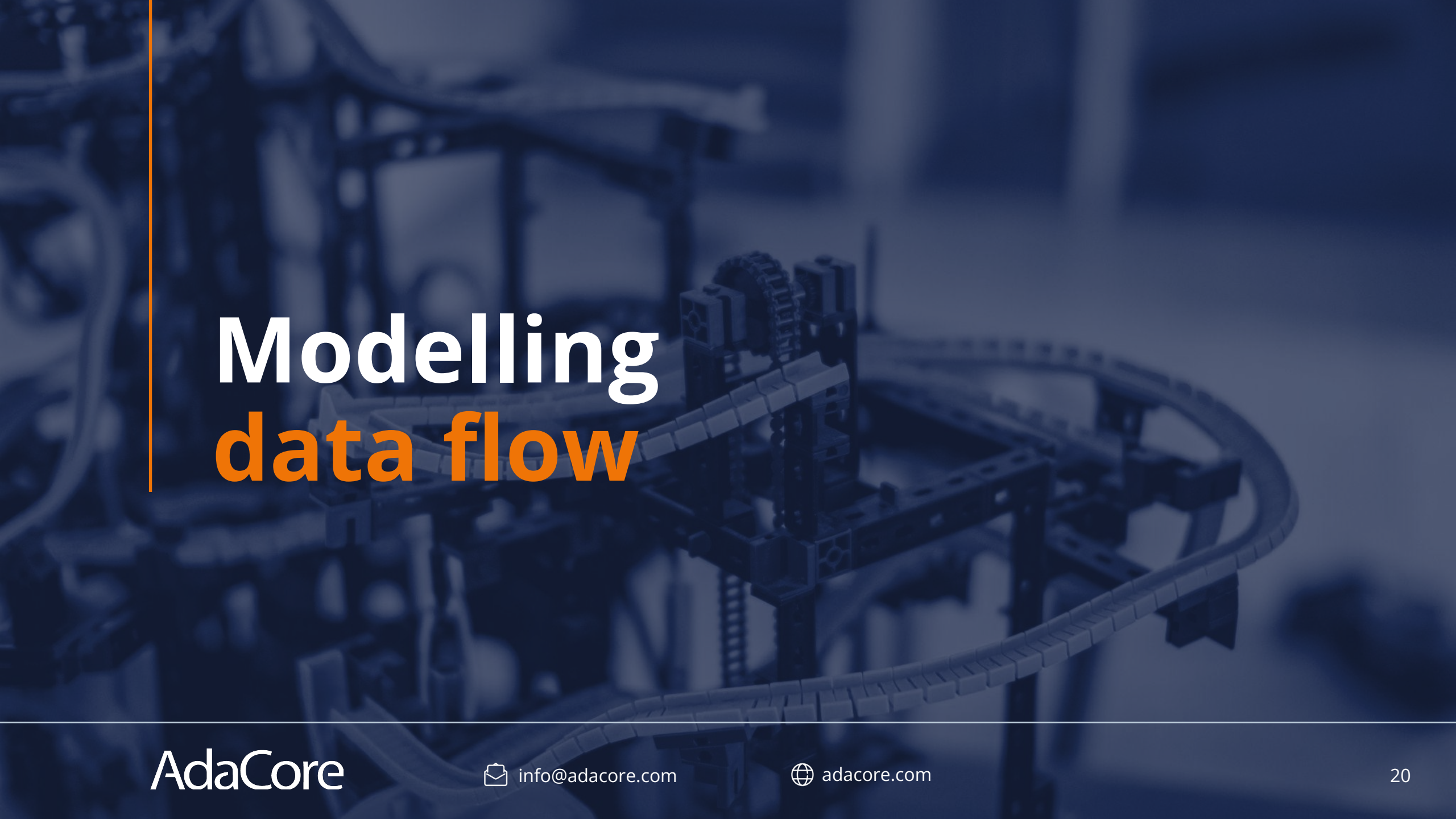
Features outside the SPARK subset

SPARK

Additional aspects to further aid static analysis

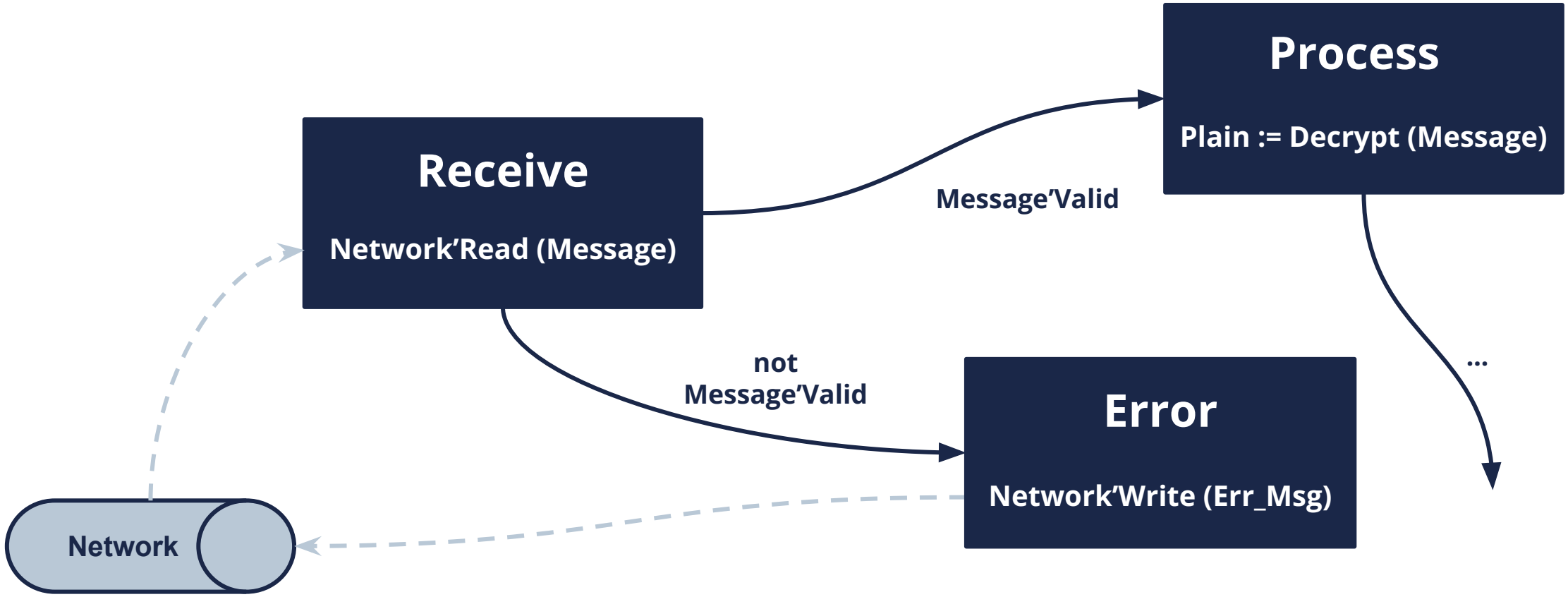
learn.adacore.com

Guarantees from program verification



Modelling data flow

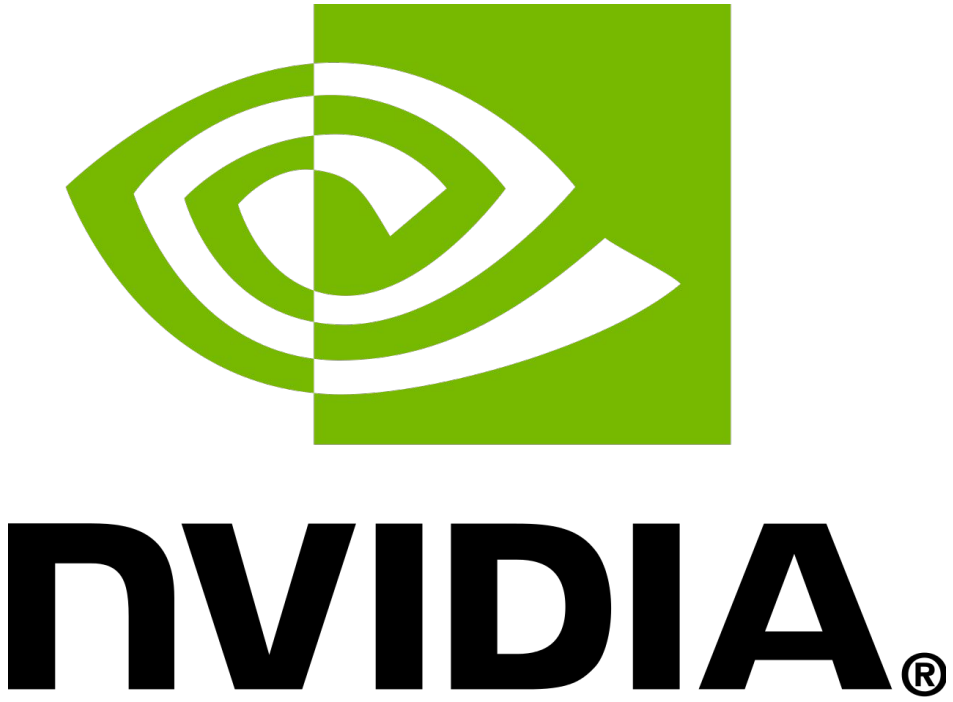
Interacting FSMs with context



Guarantees from session verification

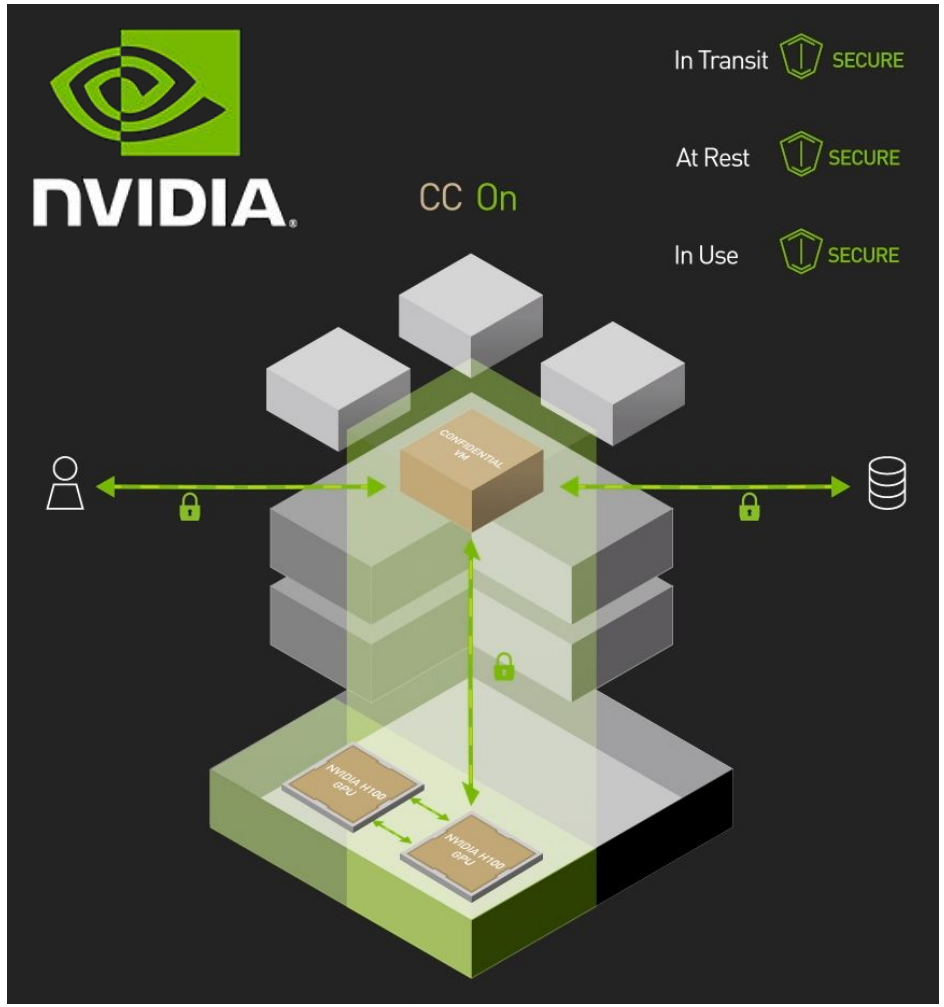
Hardening NVIDIA's Confidential Computing

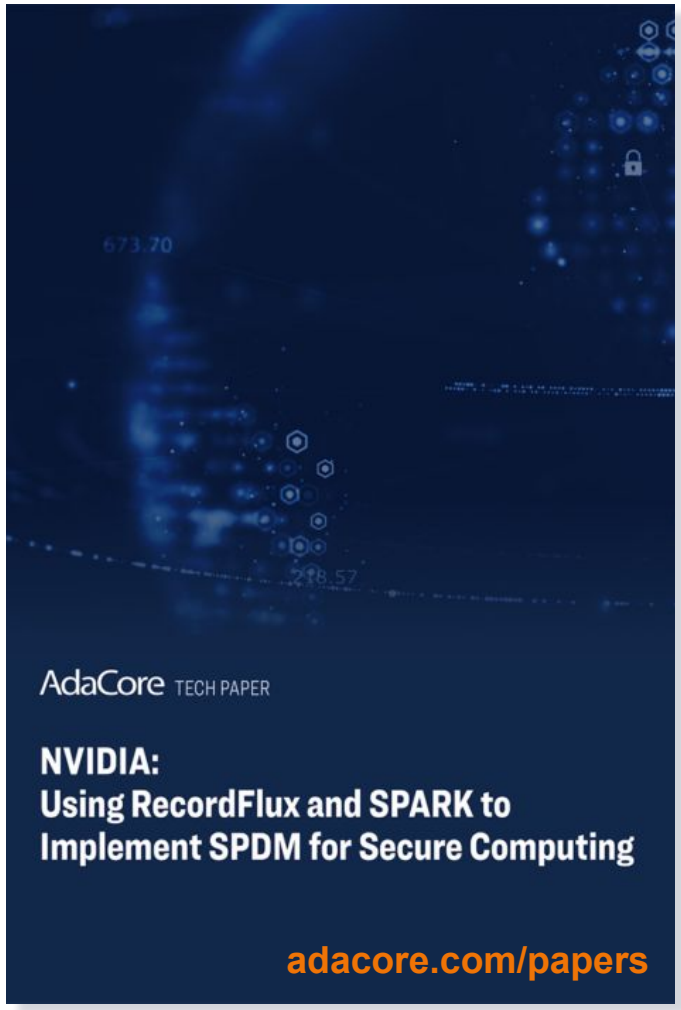
NVIDIA adopts SPARK for critical firmware



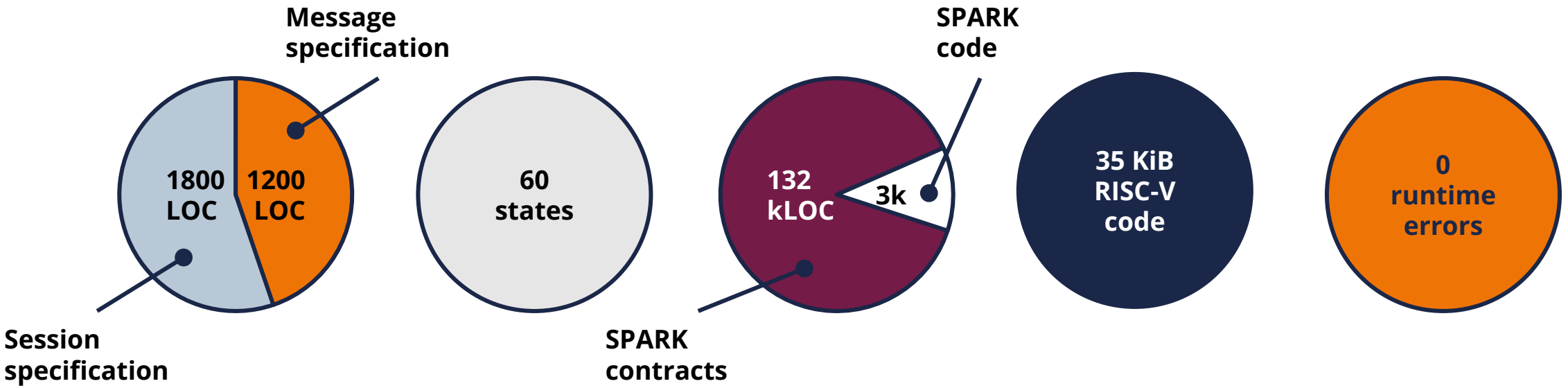
adacore.com/nvidia

Security Protocol and Data Model (SPDM)





SPDM formalization with RecordFlux



RecordFlux/SPARK SPDM implementation:
github.com/AdaCore/spdm-recordflux

The Future

adacore.com/recordflux

Tobias Reiher
reiher@adacore.com