SECOMP: Formally Secure Compilation of Compartmentalized C Programs

Cătălin Hrițcu, MPI-SP, Bochum

Joint work with
Carmine Abate, Sven Argo, **Arthur Azevedo de Amorim**, **Roberto Blanco**, **Adrien Durier**, Akram El-Korashy, **Ana Nora Evans, Guglielmo Fachini**, Deepak Garg, **Aïna Linn Georges, Théo Laurent**, Benjamin Pierce, **Marco Stronati**, **Jérémy Thibault**, Andrew Tolmach, ...

In part supported by ERC Starting Grant SECOMP

Secure Compilation of Vulnerable Source Programs

Secure Compilation of Vulnerable Source Programs

- **Insecure languages like C enable devastating vulnerabilities**

Secure Compilation of Vulnerable Source Programs

- Insecure languages like C enable devastating vulnerabilities
- Mitigate vulnerabilities by compartmentalizing the program

Compartment 1

Compartment 2

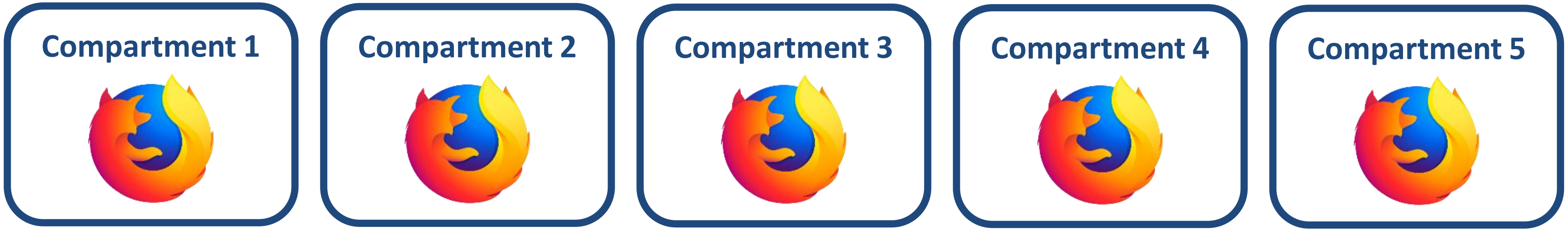
Compartment 3

Compartment 4

Compartment 5

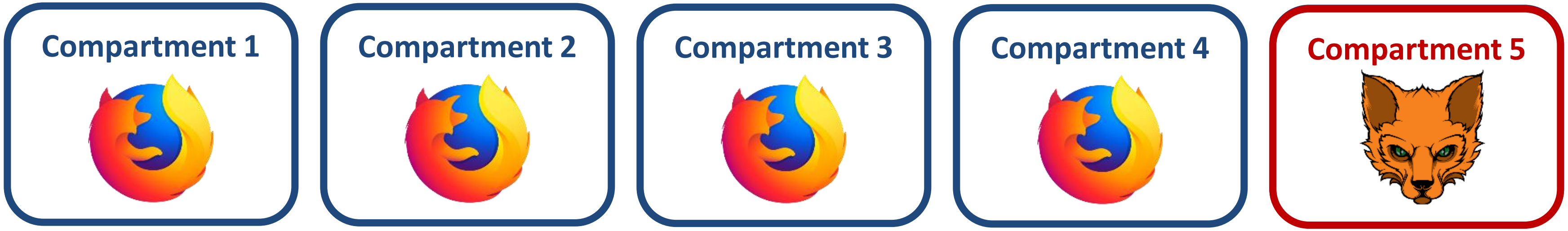
Secure Compilation of Vulnerable Source Programs

- **Insecure languages like C enable devastating vulnerabilities**
- **Mitigate vulnerabilities by compartmentalizing the program**
- **We don't know which compartments will be compromised**



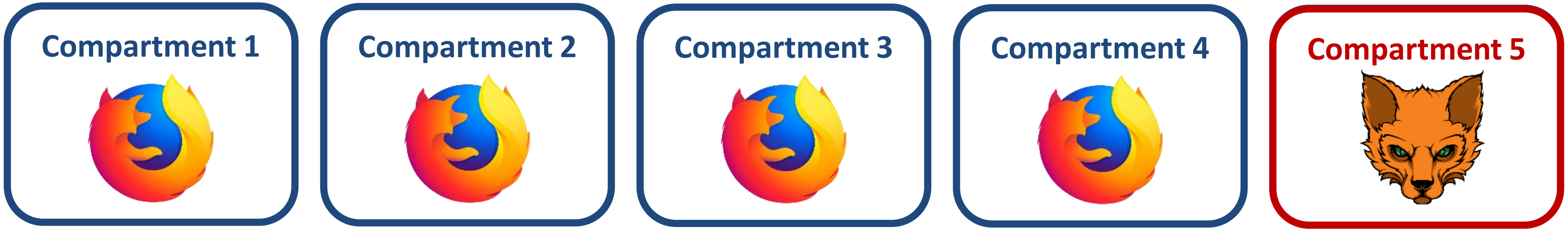
Secure Compilation of Vulnerable Source Programs

- Insecure languages like C enable devastating vulnerabilities
- Mitigate vulnerabilities by compartmentalizing the program
- We don't know which compartments will be compromised



Secure Compilation of Vulnerable Source Programs

- **Insecure languages like C enable devastating vulnerabilities**
- **Mitigate vulnerabilities by compartmentalizing the program**
- **We don't know which compartments will be compromised**
  - protect vulnerable C compartments from each other



Compartment 1

Compartment 2

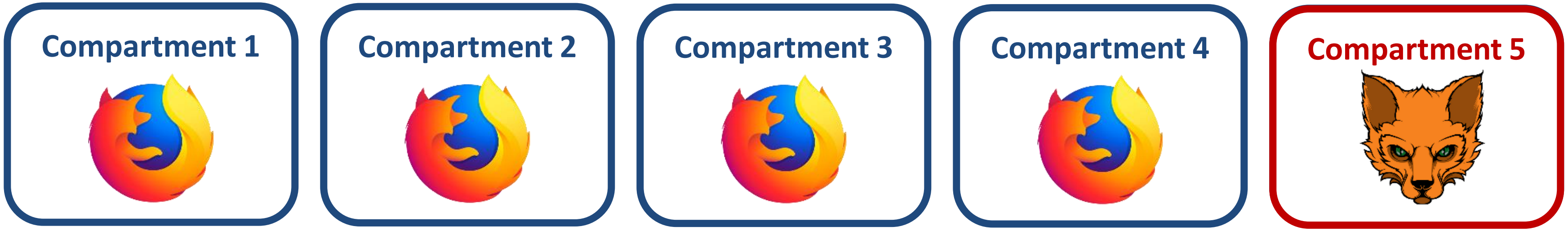
Compartment 3

Compartment 4

Compartment 5

Secure Compilation of Vulnerable Source Programs

- Insecure languages like C enable **devastating vulnerabilities**
- **Mitigate vulnerabilities by compartmentalizing the program**
- **We don't know which compartments will be compromised**
  - protect vulnerable C compartments from each other
- **We don't know when a compartment will be compromised**



Compartment 1

Compartment 2

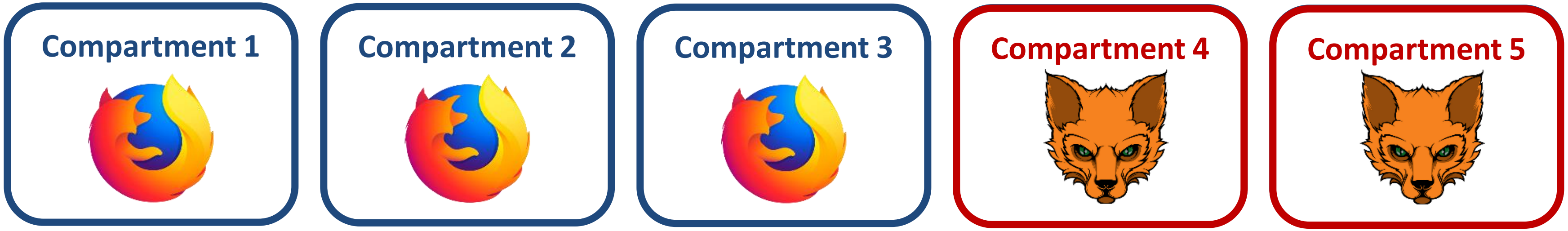
Compartment 3

Compartment 4

Compartment 5

Secure Compilation of Vulnerable Source Programs

- Insecure languages like C enable devastating vulnerabilities
- Mitigate vulnerabilities by compartmentalizing the program
- We don't know which compartments will be compromised
  - protect vulnerable C compartments from each other
- We don't know when a compartment will be compromised



Compartment 1

Compartment 2

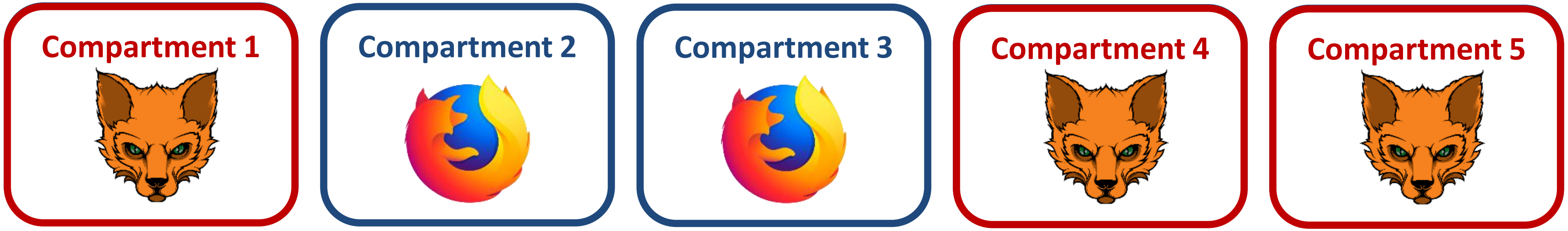
Compartment 3

Compartment 4

Compartment 5

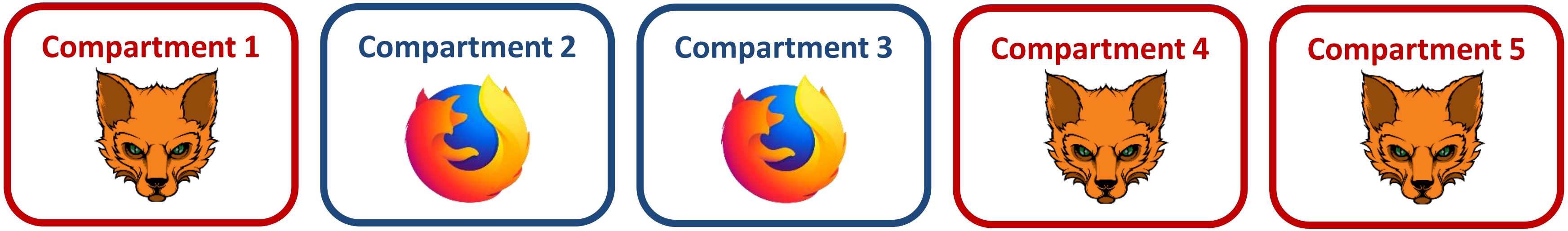
Secure Compilation of Vulnerable Source Programs

- **Insecure languages like C enable devastating vulnerabilities**
- **Mitigate vulnerabilities by compartmentalizing the program**
- **We don't know which compartments will be compromised**
  - protect vulnerable C compartments from each other
- **We don't know when a compartment will be compromised**



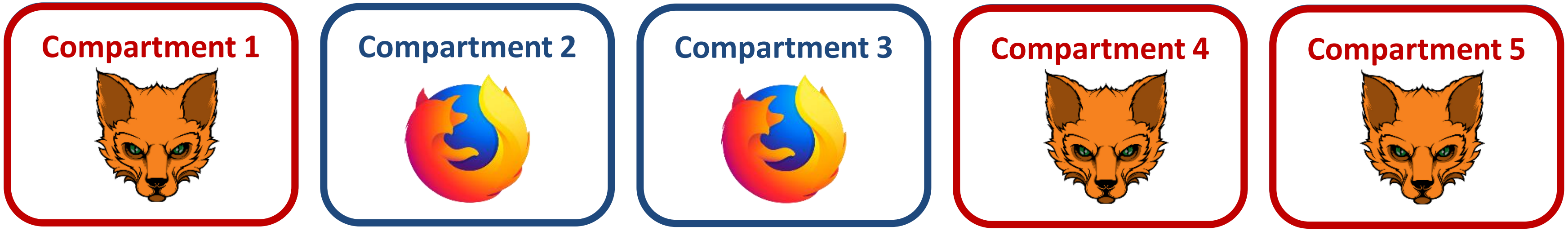
Secure Compilation of Vulnerable Source Programs

- **Insecure languages like C enable devastating vulnerabilities**
- **Mitigate vulnerabilities by compartmentalizing the program**
- **We don't know which compartments will be compromised**
  - protect vulnerable C compartments from each other
- **We don't know when a compartment will be compromised**
  - every compartment should receive protection until compromised



Secure Compilation of Vulnerable Source Programs

- **Insecure languages like C enable devastating vulnerabilities**
- **Mitigate vulnerabilities by compartmentalizing the program**
- **We don't know which compartments will be compromised**
  - protect vulnerable C compartments from each other
- **We don't know when a compartment will be compromised**
  - every compartment should receive protection until compromised
- Formalized this as a **variant of robust safety preservation** [CCS'18]



Compartment 1

Compartment 2

Compartment 3

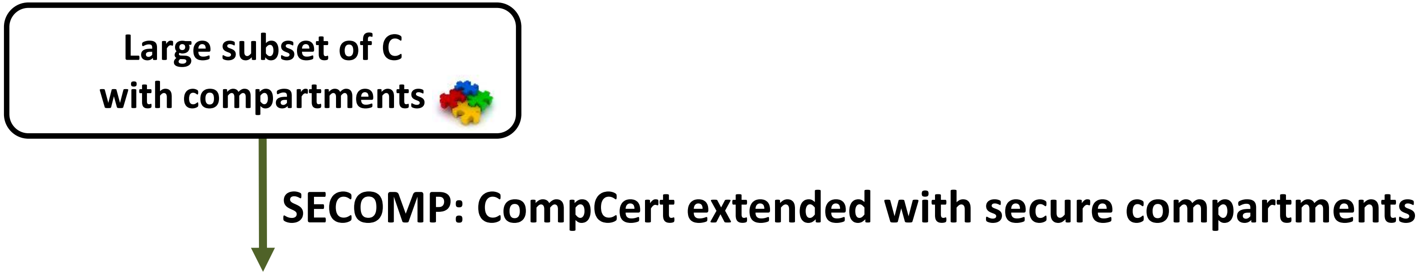
Compartment 4

Compartment 5

Security Enforcement

Large subset of C
with compartments 🧩

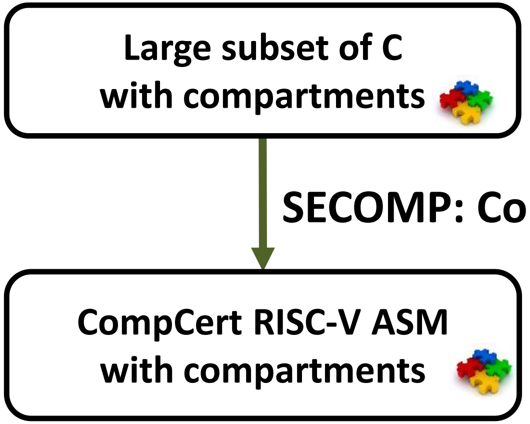
Security Enforcement



Large subset of C
with compartments

SECOMP: CompCert extended with secure compartments

Security Enforcement



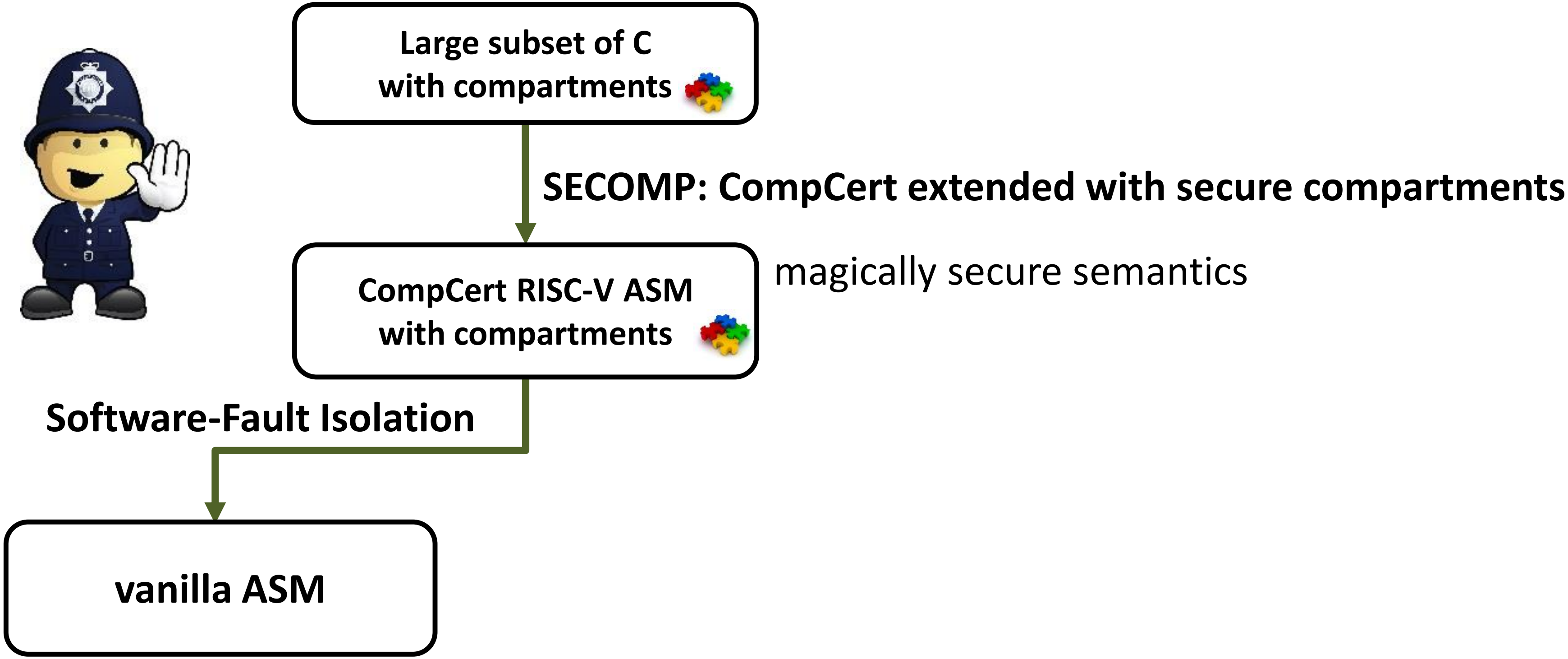
Large subset of C
with compartments

SECOMP: CompCert extended with secure compartments

CompCert RISC-V ASM
with compartments

magically secure semantics

Security Enforcement



Large subset of C with compartments

SECOMP: CompCert extended with secure compartments

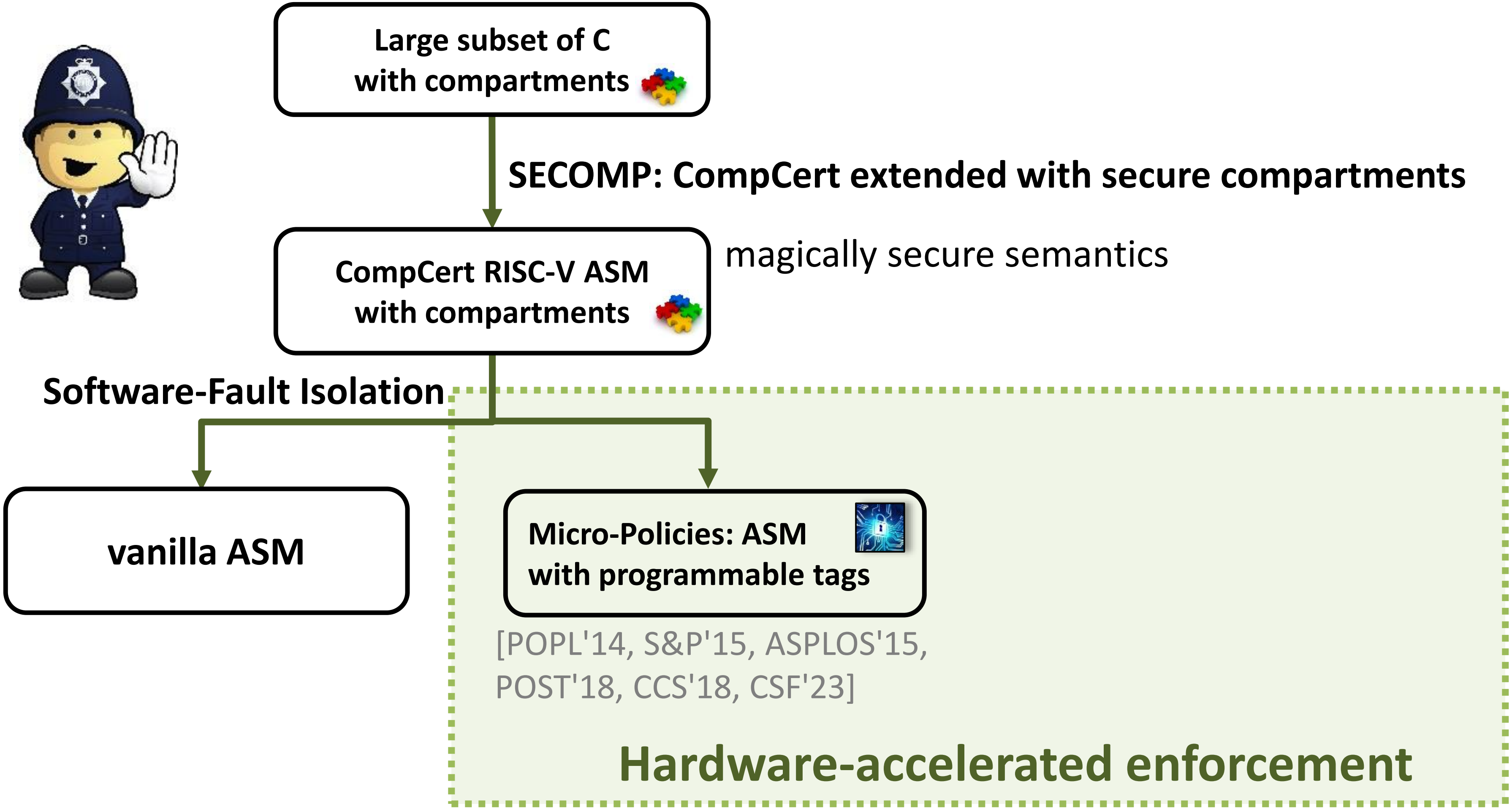
CompCert RISC-V ASM with compartments

magically secure semantics

Software-Fault Isolation

vanilla ASM

Security Enforcement



Large subset of C
with compartments

SECOMP: CompCert extended with secure compartments

CompCert RISC-V ASM
with compartments

magically secure semantics

Software-Fault Isolation

vanilla ASM

Micro-Policies: ASM
with programmable tags

[POPL'14, S&P'15, ASPLOS'15,
POST'18, CCS'18, CSF'23]

Hardware-accelerated enforcement

Security Enforcement

Large subset of C with compartments

SECOMP: CompCert extended with secure compartments

CompCert RISC-V ASM with compartments

magically secure semantics

Software-Fault Isolation

vanilla ASM

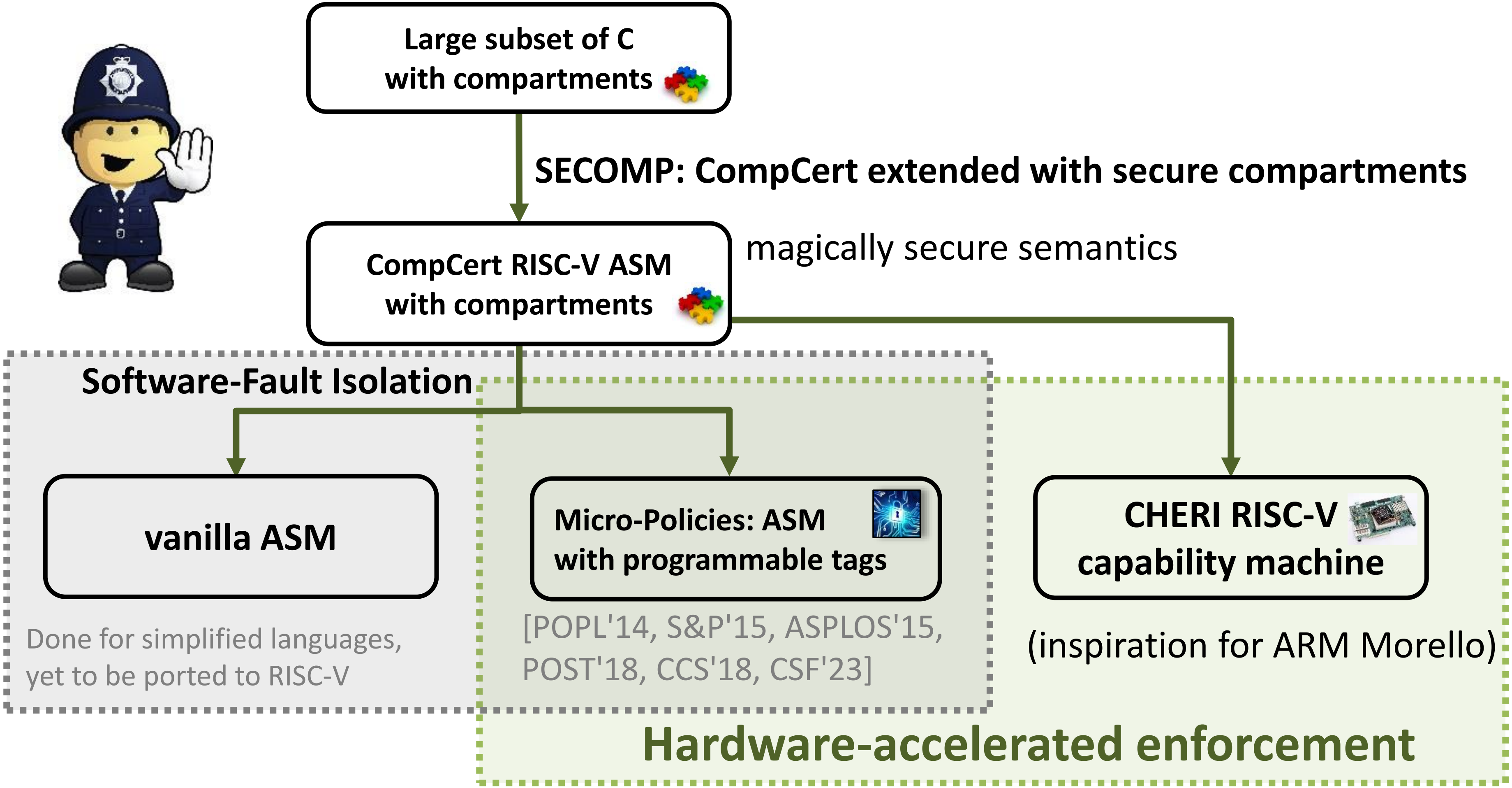
Done for simplified languages, yet to be ported to RISC-V

Micro-Policies: ASM with programmable tags

[POPL'14, S&P'15, ASPLOS'15, POST'18, CCS'18, CSF'23]

Hardware-accelerated enforcement

Security Enforcement



Large subset of C with compartments

SECOMP: CompCert extended with secure compartments

CompCert RISC-V ASM with compartments

magically secure semantics

Software-Fault Isolation

vanilla ASM

Done for simplified languages, yet to be ported to RISC-V

Micro-Policies: ASM with programmable tags

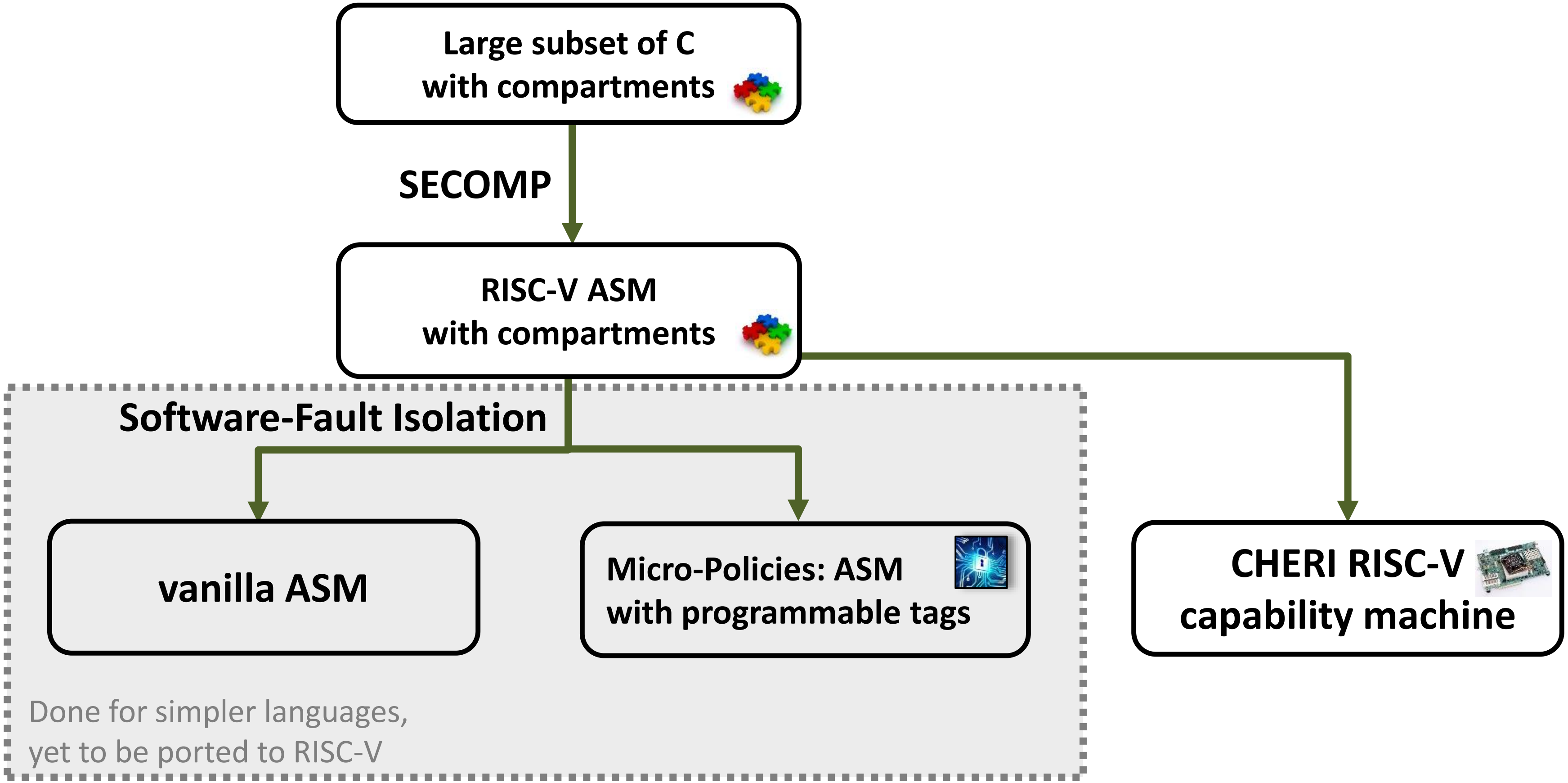
[POPL'14, S&P'15, ASPLOS'15, POST'18, CCS'18, CSF'23]

CHERI RISC-V capability machine

(inspiration for ARM Morello)

Hardware-accelerated enforcement

Secure Compilation Proofs in Coq



Large subset of C with compartments

SECOMP

RISC-V ASM with compartments

Software-Fault Isolation

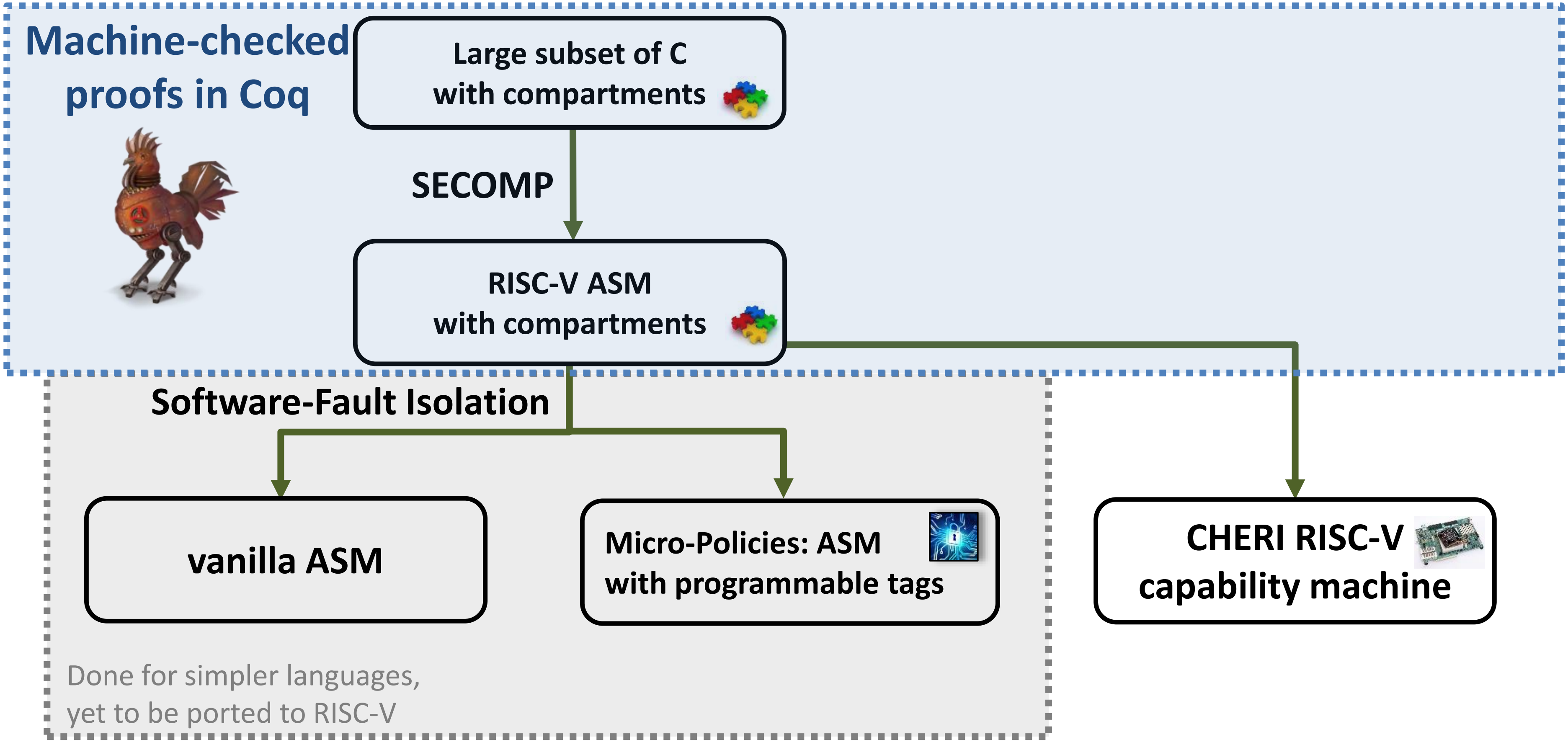
vanilla ASM

Micro-Policies: ASM with programmable tags

CHERI RISC-V capability machine

Done for simpler languages, yet to be ported to RISC-V

Secure Compilation Proofs in Coq



Machine-checked proofs in Coq

Large subset of C with compartments

SECOMP

RISC-V ASM with compartments

Software-Fault Isolation

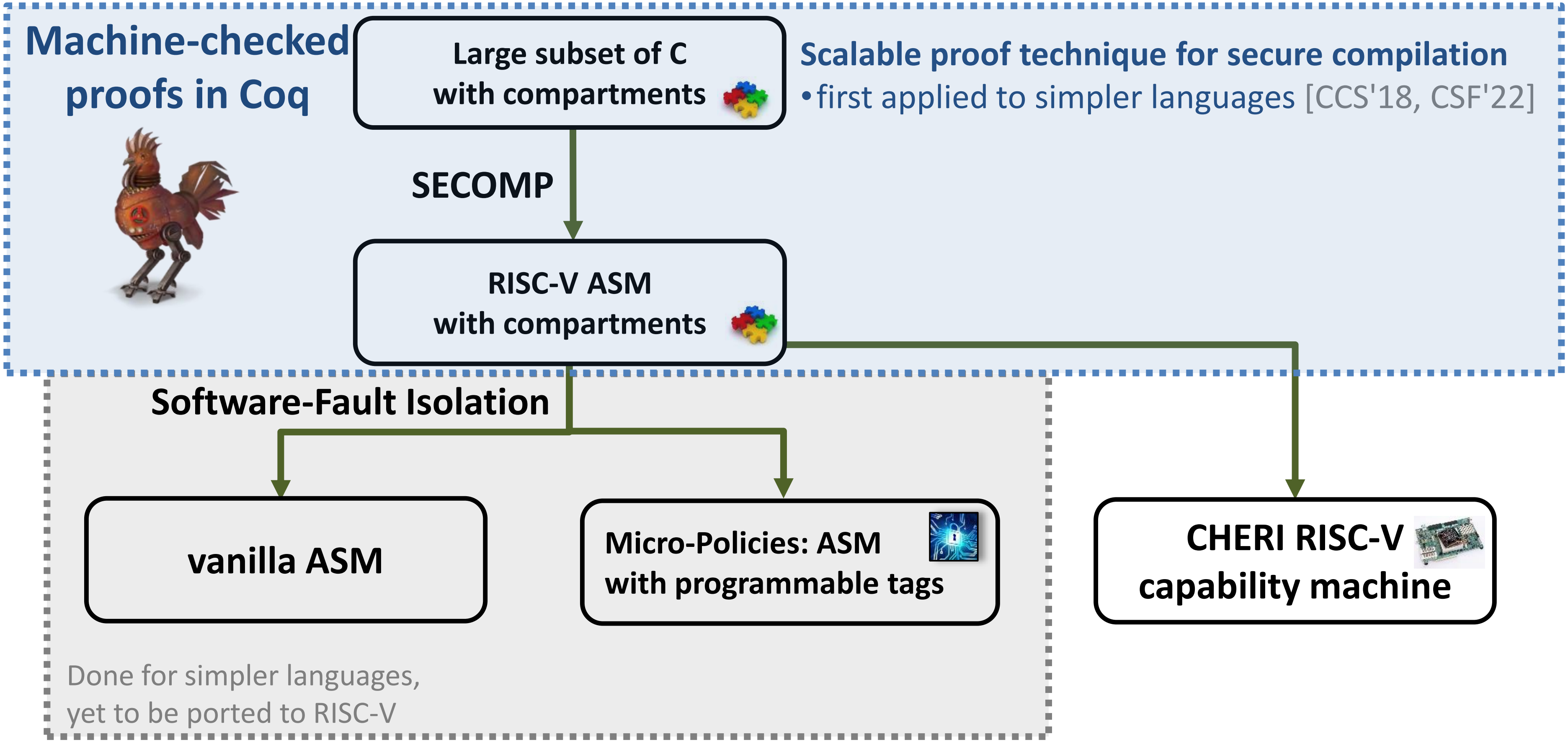
vanilla ASM

Micro-Policies: ASM with programmable tags

CHERI RISC-V capability machine

Done for simpler languages, yet to be ported to RISC-V

Secure Compilation Proofs in Coq



Machine-checked proofs in Coq

Large subset of C with compartments

SECOMP

RISC-V ASM with compartments

Scalable proof technique for secure compilation

- first applied to simpler languages [CCS'18, CSF'22]

Software-Fault Isolation

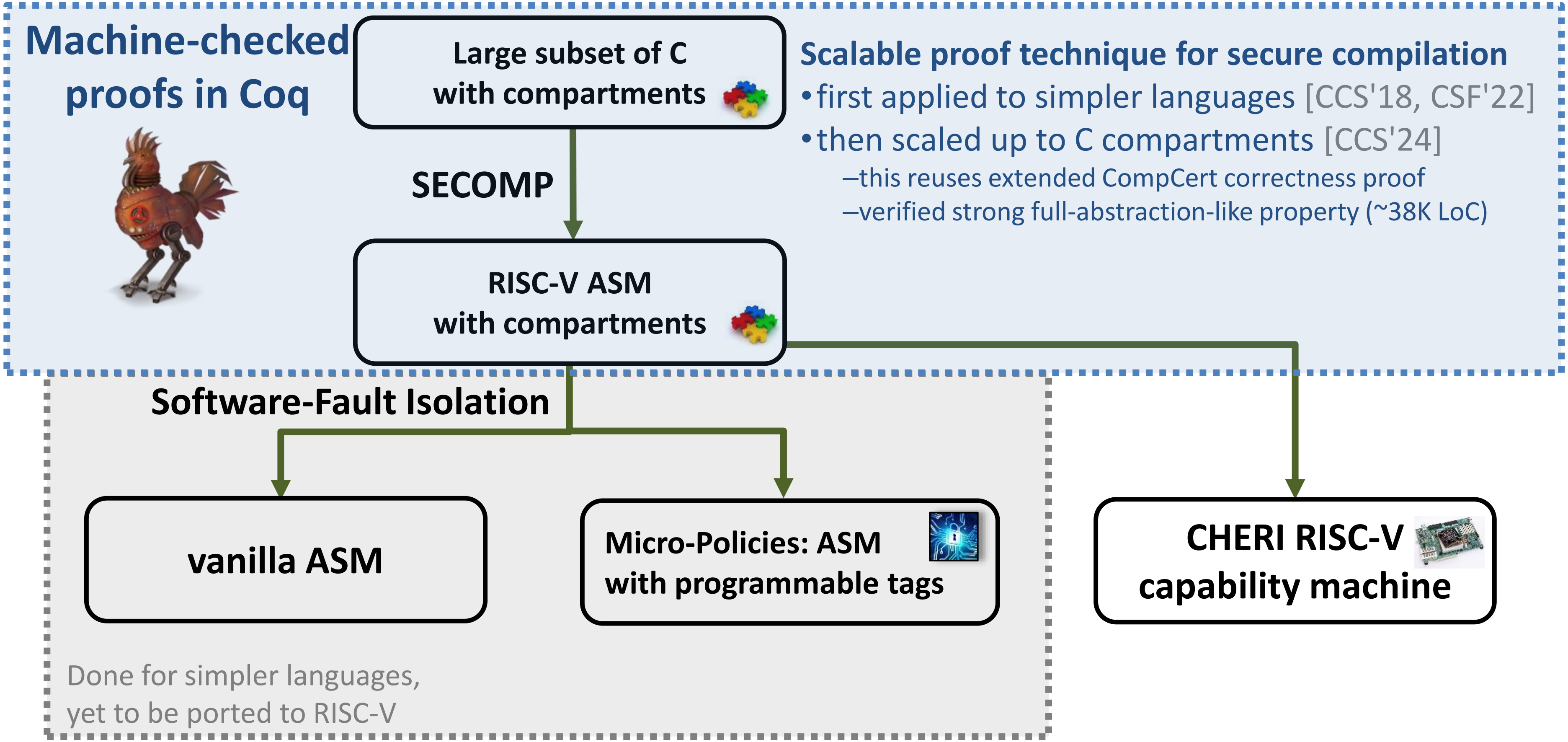
vanilla ASM

Micro-Policies: ASM with programmable tags

CHERI RISC-V capability machine

Done for simpler languages,
yet to be ported to RISC-V

Secure Compilation Proofs in Coq



Machine-checked proofs in Coq

Large subset of C with compartments

SECOMP

RISC-V ASM with compartments

Scalable proof technique for secure compilation

- first applied to simpler languages [CCS'18, CSF'22]
- then scaled up to C compartments [CCS'24]
  - this reuses extended CompCert correctness proof
  - verified strong full-abstraction-like property (~38K LoC)

Software-Fault Isolation

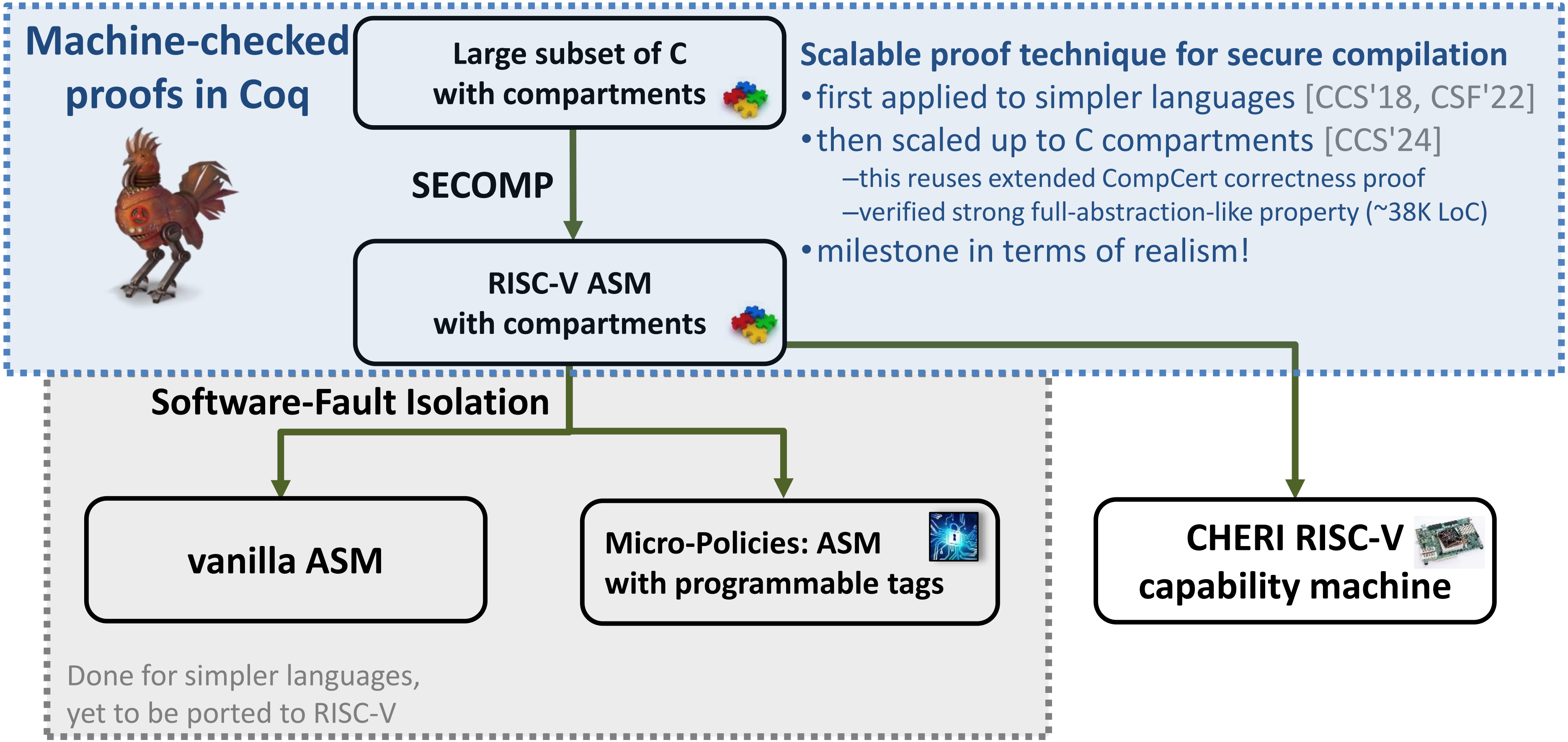
vanilla ASM

Micro-Policies: ASM with programmable tags

CHERI RISC-V capability machine

Done for simpler languages,
yet to be ported to RISC-V

Secure Compilation Proofs in Coq



Machine-checked proofs in Coq

Large subset of C with compartments

SECOMP

RISC-V ASM with compartments

Software-Fault Isolation

vanilla ASM

Micro-Policies: ASM with programmable tags

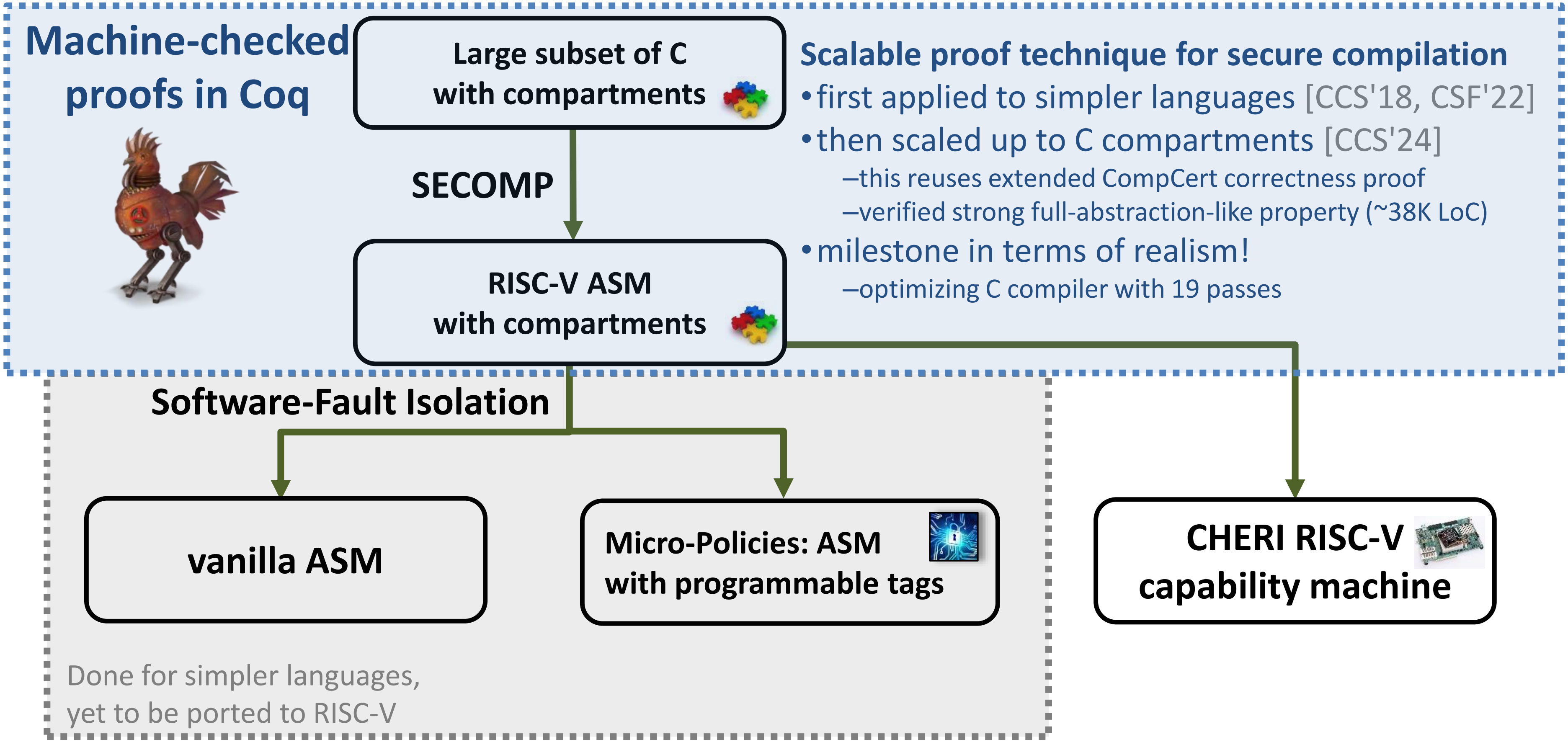
CHERI RISC-V capability machine

Done for simpler languages, yet to be ported to RISC-V

Scalable proof technique for secure compilation

- first applied to simpler languages [CCS'18, CSF'22]
- then scaled up to C compartments [CCS'24]
  - this reuses extended CompCert correctness proof
  - verified strong full-abstraction-like property (~38K LoC)
- milestone in terms of realism!

Secure Compilation Proofs in Coq



Machine-checked proofs in Coq

Large subset of C with compartments

SECOMP

RISC-V ASM with compartments

Scalable proof technique for secure compilation

- first applied to simpler languages [CCS'18, CSF'22]
- then scaled up to C compartments [CCS'24]
  - this reuses extended CompCert correctness proof
  - verified strong full-abstraction-like property (~38K LoC)
- milestone in terms of realism!
  - optimizing C compiler with 19 passes

Software-Fault Isolation

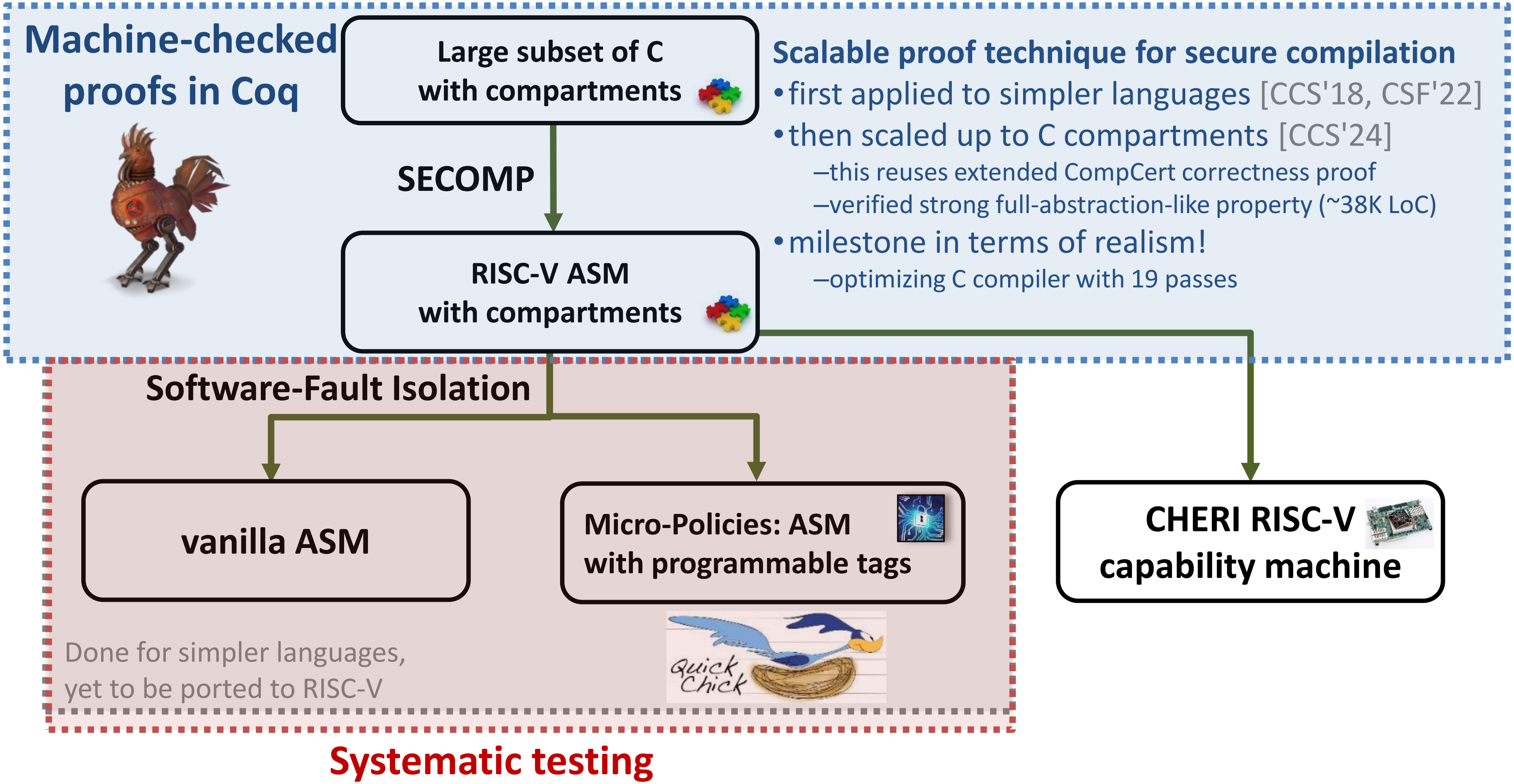
vanilla ASM

Micro-Policies: ASM with programmable tags

CHERI RISC-V capability machine

Done for simpler languages, yet to be ported to RISC-V

Secure Compilation Proofs in Coq



Machine-checked proofs in Coq

Large subset of C with compartments

SECOMP

RISC-V ASM with compartments

Scalable proof technique for secure compilation

- first applied to simpler languages [CCS'18, CSF'22]
- then scaled up to C compartments [CCS'24]
  - –this reuses extended CompCert correctness proof
  - –verified strong full-abstraction-like property (~38K LoC)
- milestone in terms of realism!
  - –optimizing C compiler with 19 passes

Software-Fault Isolation

vanilla ASM

Micro-Policies: ASM with programmable tags

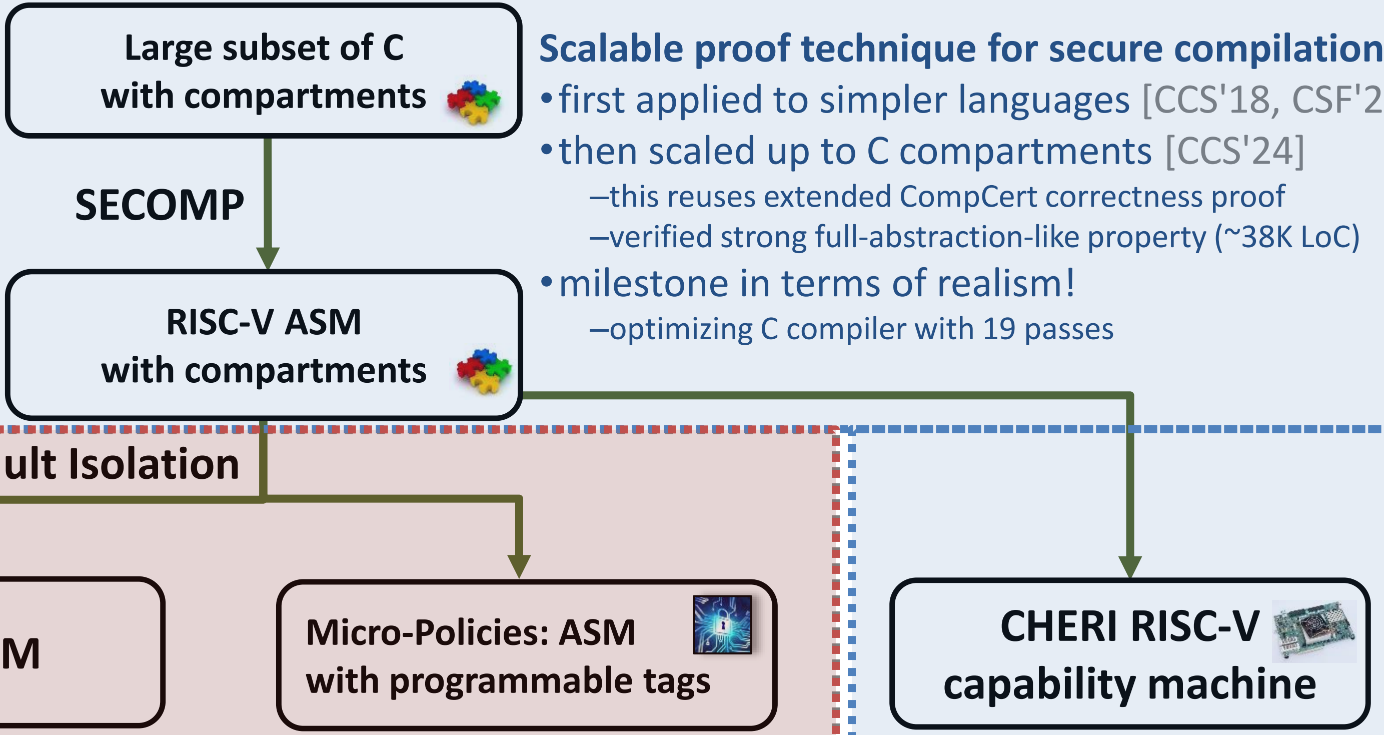
CHERI RISC-V capability machine

Done for simpler languages, yet to be ported to RISC-V

Systematic testing

Secure Compilation Proofs in Coq

Machine-checked proofs in Coq



Large subset of C with compartments

SECOMP

RISC-V ASM with compartments

Scalable proof technique for secure compilation

- first applied to simpler languages [CCS'18, CSF'22]
- then scaled up to C compartments [CCS'24]
  - this reuses extended CompCert correctness proof
  - verified strong full-abstraction-like property (~38K LoC)
- milestone in terms of realism!
  - optimizing C compiler with 19 passes

Software-Fault Isolation

vanilla ASM

Micro-Policies: ASM with programmable tags

CHERI RISC-V capability machine

Done for simpler languages, yet to be ported to RISC-V

Big verification challenge for the future

Systematic testing

Future Plans on Formally Secure Compilation

Verify capability backend

Future Plans on Formally Secure Compilation

Better Proof Techniques

Verify capability backend

Future Plans on Formally Secure Compilation

Better Proof Techniques

Capability passing

Verify capability backend

Future Plans on Formally Secure Compilation

Preserve data confidentiality

Better Proof Techniques

Capability passing

Verify capability backend

Future Plans on Formally Secure Compilation

Stronger Security Goals

Preserve data confidentiality

Better Proof Techniques

Capability passing

Verify capability backend

Future Plans on Formally Secure Compilation

SPECTRE

Stronger Security Goals

Preserve data confidentiality
against micro-architectural side-channel attacks,
for compartmentalized programs in F*, C, or Wasm

Better Proof Techniques

Capability passing

Verify capability backend

Future Plans on Formally Secure Compilation

Stronger Security Goals

Preserve data confidentiality against micro-architectural side-channel attacks, for compartmentalized programs in F*, C, or Wasm

Realistic Enforcement

Capability passing

Better Proof Techniques

Verify capability backend

Future Plans on Formally Secure Compilation

Stronger Security Goals

Preserve data confidentiality
against micro-architectural side-channel attacks,
for compartmentalized programs in F*, C, or Wasm

Realistic Enforcement

ARM Morello
capability machine

Capability passing

Better Proof Techniques

Verify capability backend