Towards a Verified Component Platform

Matthew Fernandez, Ihor Kuz, Gerwin Klein, June Andronick

November 2013
Trustworthy Systems
Trustworthy Systems

- Development processes
- Testing
- Configuration management
- Certification
- Formal verification
The L4.verified Project

- **Specification**: 5,000 LOC
- **C Code**: 10,000 LOC
- **Proof**: 200,000 LOC
- **25 person years**
The L4.verified Project

25 person years

Specification: 5,000 LOC

C Code: 10,000 LOC

1,000,000 LOC

NICTA
Component-Based Development
Component-Based Development
Component-Based Development

- kernel
- component platform
- components
Component-Based Development

seL4

+ kernel

+ component platform

+ components
Component-Based Development

seL4

kernel

+ component platform

user-provided

components
Component-Based Development

seL4 + CAmkES + user-provided components

kernel + component platform
End-to-end Guarantees

architecture description

component code
End-to-end Guarantees
End-to-end Guarantees
End-to-end Guarantees

architecture semantics

architecture description

glue code

component code

userspace image
End-to-end Guarantees

architecture semantics

architecture description

glue semantics

glue code

trusted component specification

component code

userspace image
End-to-end Guarantees

architecture semantics

system semantics

trusted component specification

architecture description

glue semantics

component code

userspace image

glue code

Copyright NICTA 2013
End-to-end Guarantees

system semantics

architecture semantics  glue semantics  trusted component specification

architecture description  glue code  component code

userspace image
End-to-end Guarantees

NICTA

architecture semantics → glue semantics → trusted component specification
architecture description → glue code → component code

system semantics

userspace image
End-to-end Guarantees

system semantics

architecture semantics

glue semantics

trusted component specification

userspace image
End-to-end Guarantees

- Architecture semantics
- Glue semantics
- Trusted component specification

System semantics

Userspace image
CAmkES

- Component platform for seL4
- Abstractions for procedures, events, shared memory
- Limited to static architectures

![Diagram of A and B connected by f]
CAmkES

- Component platform for seL4
- Abstractions for procedures, events, shared memory
- Limited to static architectures
CAmkES

- Component platform for seL4
- Abstractions for procedures, events, shared memory
- Limited to static architectures

```
A:
  x = 1
  y = f(x)
  print y

B:
  f(v) {
    return v + 1
  }

f(w) {
  marshal w
  send()
  wait()
  unmarshal z
  return z

while (true) {
  wait()
  unmarshal p
  r = f(p)
  marshal r
  send()
}```
CAmkES

- Component platform for seL4
- Abstractions for procedures, events, shared memory
- Limited to static architectures

A:
  x = 1
  y = f(x)
  print y

B:
  f(v) {
    return v + 1
  }

f(w) {
  marshal w
  send()
  wait()
  unmarshal z
  return z
}

while (true) {
  wait()
  unmarshal p
  r = f(p)
  marshal r
  send()
}
CAmkES

- Component platform for seL4
- Abstractions for procedures, events, shared memory
- Limited to static architectures
CAmkES

- Component platform for seL4
- Abstractions for procedures, events, shared memory
- Limited to static architectures
Architecture Semantics

composition {
    component Client A;
    component Server B;

    connection RPC f(from A.i, to B.j);
}
composition {
  component Client A;
  component Server B;

  connection RPC f(from A.i, to B.j);
}

definition system :: composition
where
  system ≡ {
    components = [('A', Client), ('B', Server)],
    connections = [('f', { conn_type = RPC,
                          conn_from = ('A', 'i'),
                          conn_to = ('B', 'j') })]
  }
Component Process Language

UserStep \( x = 1 \)

\[
\begin{align*}
\text{SKIP} \\
S_1 \;;\; S_2 \\
\text{IF } c \text{ THEN } S_1 \text{ ELSE } S_2 \\
\text{WHILE } c \text{ DO } s
\end{align*}
\]

\( S_1 \cup S_2 \quad \leftarrow \quad \text{Request} \quad \text{Response} \)
Component Behaviour

A \rightarrow f \rightarrow B
Component Behaviour
Component Behaviour

definition
B_untrusted :: ...

where
B_untrusted ≡
WHILE True DO (  
  UserStep
  • ArbitraryRequest f
  • ArbitraryResponse f
)
definition
  Receive_B_f :: ...
where
  Receive_B_f f embed f fproject ≡
  Response ... ;;
  f ;;
  Request ...

wait()
unmarshal p
r = f(p)
marshal r
send()
Glue Code Behaviour

definition
   Receive_B_f :: ...
where
   Receive_B_f f embed f f project ≡
       Response ... ;;
       f ;;
       Request ...

wait()
unmarshal p
r = f(p)
marshal r
send()
**Glue Code Behaviour**

```haskell
definition
  Receive_B_f :: ...
where
  Receive_B_f f embed f fproject ≡
    Response ... ;;
    f ;;
    Request ...

wait()
unmarshal p
r = f(p)
marshal r
send()
```
**Glue Code Behaviour**

```plaintext
definition
  Receive_B_f :: ...
where
  Receive_B_f fembed f fproject ≡
    Response ... ;
    f ;
    Request ...
```

```plaintext
wait()
unmarshal p
r = f(p)
marshal r
send()
```
Introduced Components

a → b
Introduced Components
Introduced Components

```
definition
  event :: ...
where
  event c ≝ WHILE True DO
    Response (λq s'. case s' of Event s →
      (case q_data q of
        Set → ...
        | Poll → ...
        | _ → {})))
```
Introduced Components

definition
  event :: ...
where
  event c ≡ WHILE True DO
    Response (λq s'. case s' of Event s →
      (case q_data q of
        Set → ... |
        Poll → ...
      ) → {}})

definition
  memory :: ...
where
  memory c ≡ WHILE True DO
    Response (λq s'. case s' of Memory s →
      (case q_data q of
        Read addr → ... |
        Write addr val → ...
      ) → {}})
Towards a Verified Component Platform
Towards a Verified Component Platform

- architecture semantics
- glue semantics
- trusted component specification

- architecture description
- glue code
- component code

- userspace image

System semantics
Towards a Verified Component Platform

- architecture semantics
- glue semantics
- trusted component specification

system semantics

userspace image