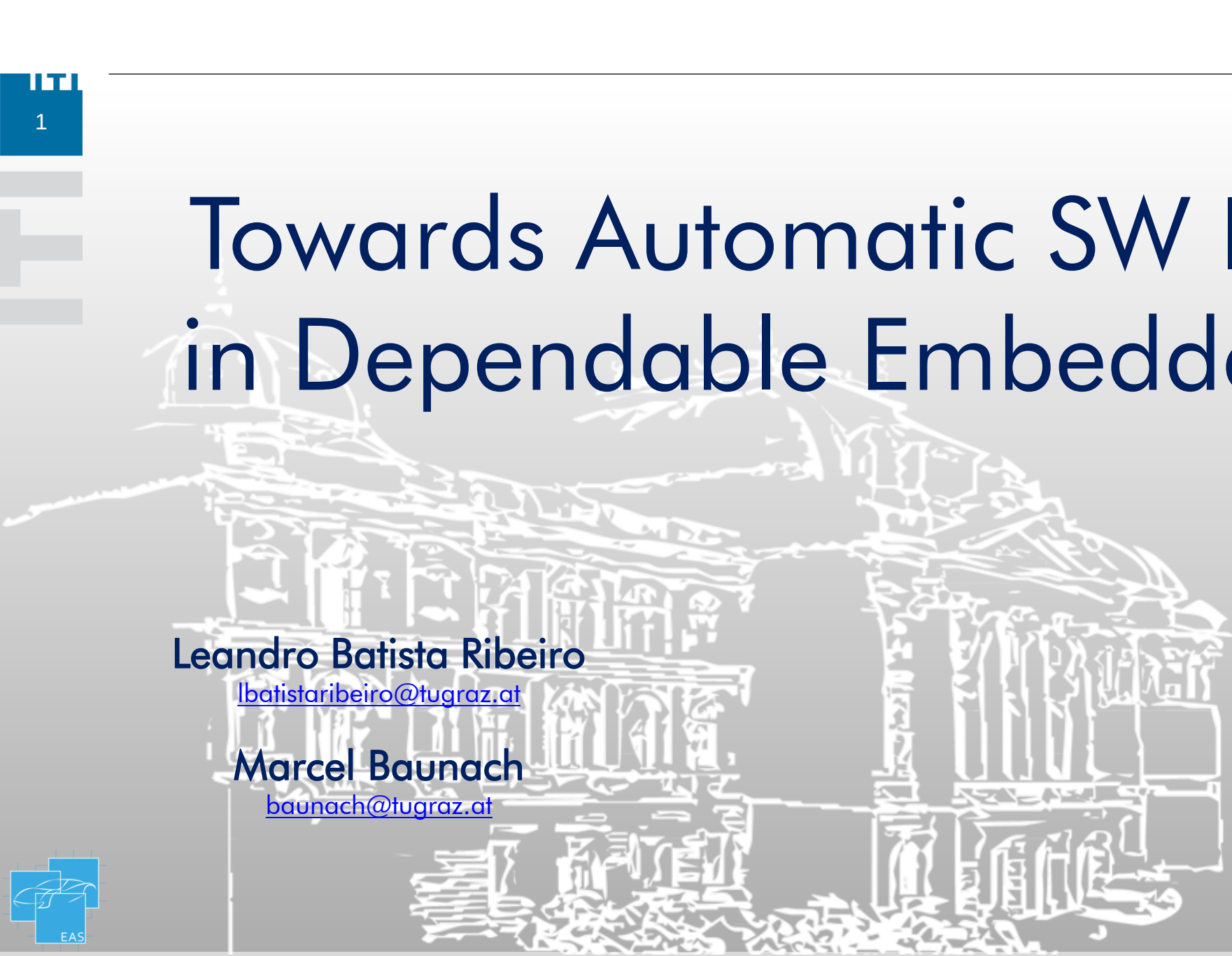
Towards Automatic SW Integration in Dependable Embedded Systems



Leandro Batista Ribeiro
lbatistaribeiro@tugraz.at

Marcel Baunach
baunach@tugraz.at

Institute of Technical Informatics
Embedded Automotive Systems Group
Graz University of Technology

Embedded SW Development

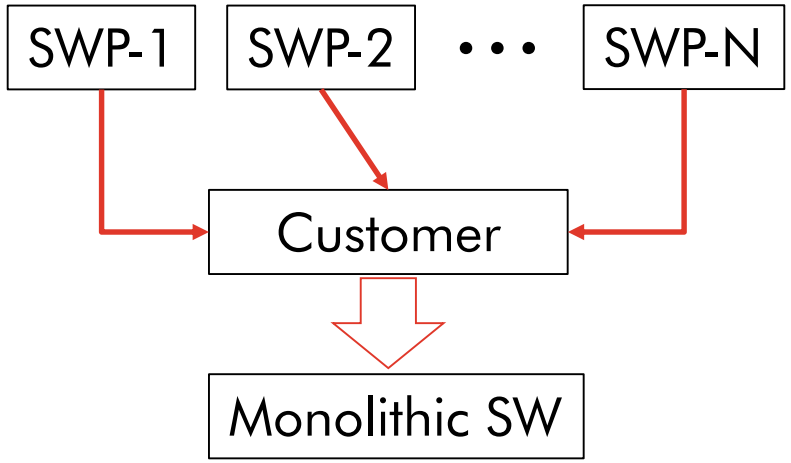
State-of-the-art

- N SW Providers (SWP)

Embedded SW Development

State-of-the-art

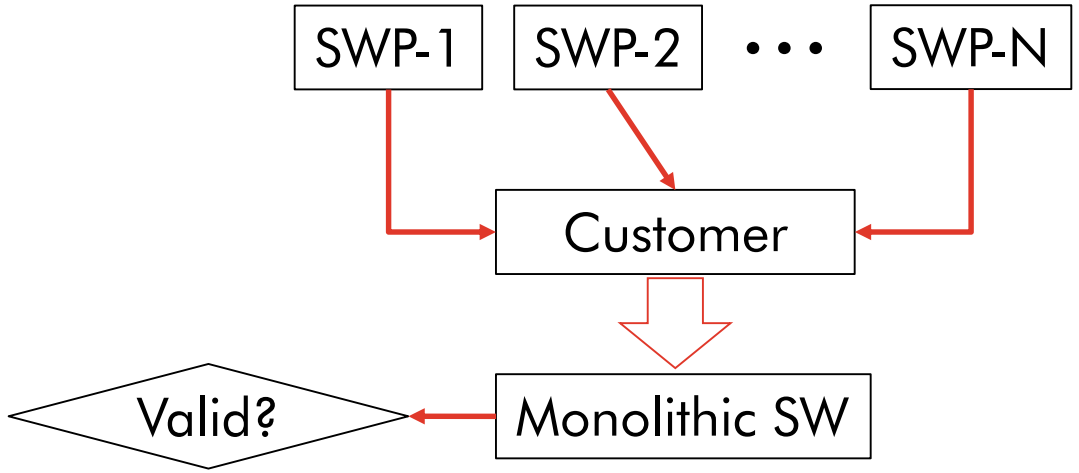
- N SW Providers (SWP)
- Customer is SW Integrator



Embedded SW Development

State-of-the-art

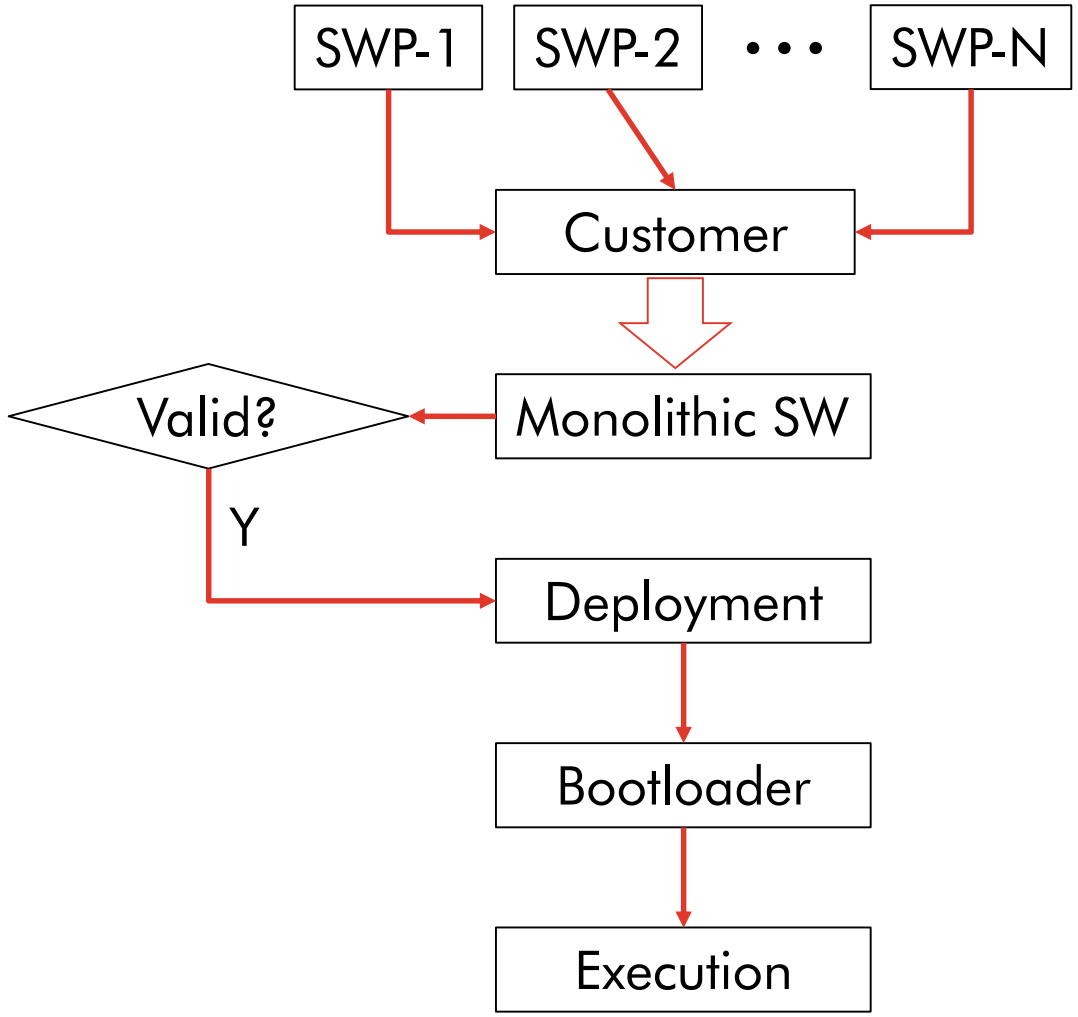
- N SW Providers (SWP)
- Customer is SW Integrator
  - Human-Supervised



Embedded SW Development

State-of-the-art

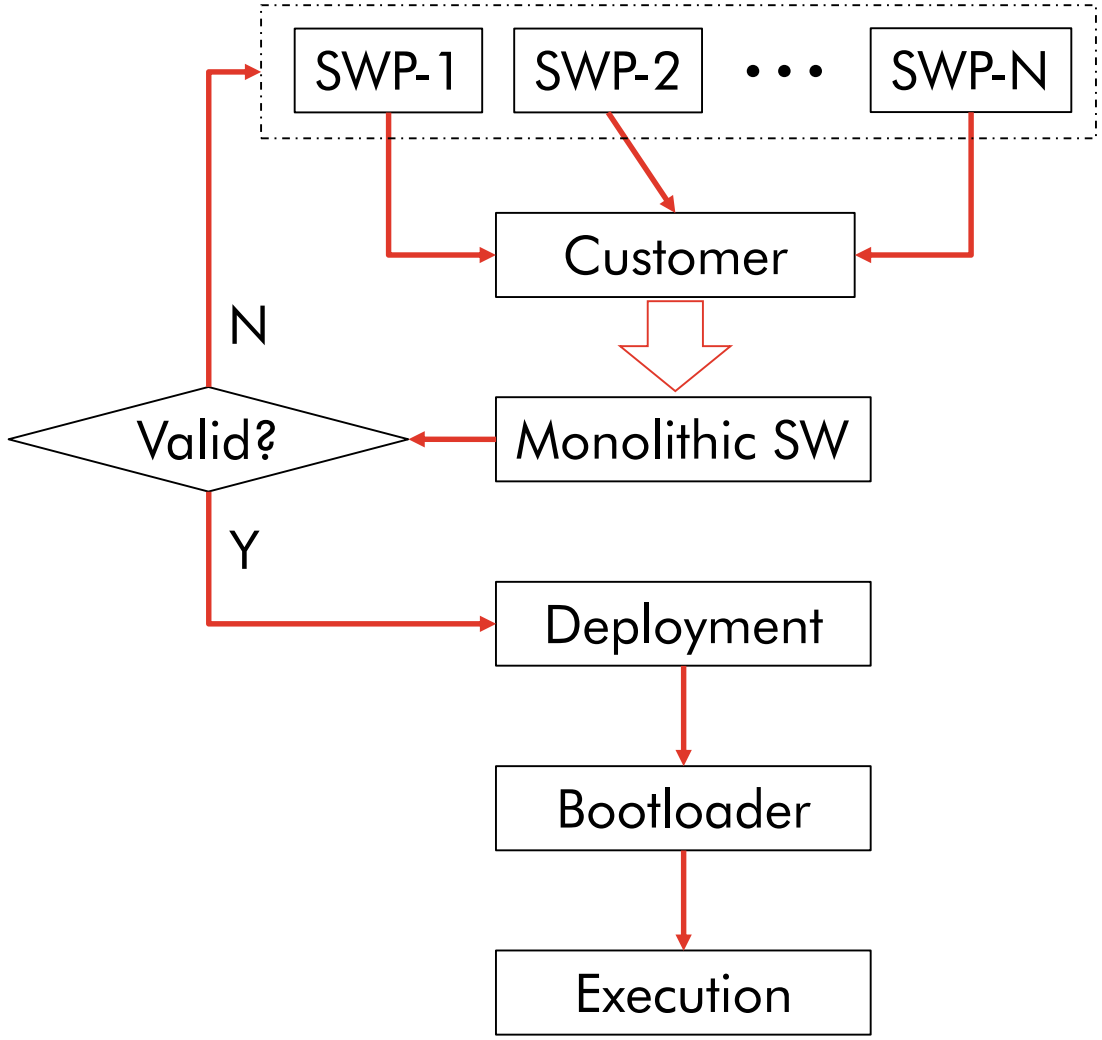
- N SW Providers (SWP)
- Customer is SW Integrator
  - Human-Supervised
- Monolithic SW is deployed
  - Downtime



Embedded SW Development

State-of-the-art

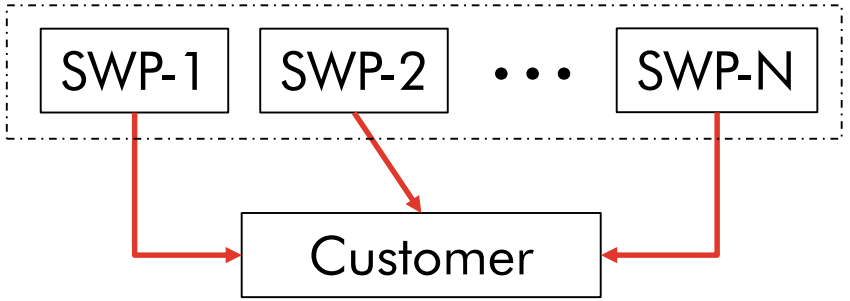
- N SW Providers (SWP)
- Customer is SW Integrator
  - Human-Supervised
- Monolithic SW is deployed
  - Downtime



Embedded SW Development

Vision

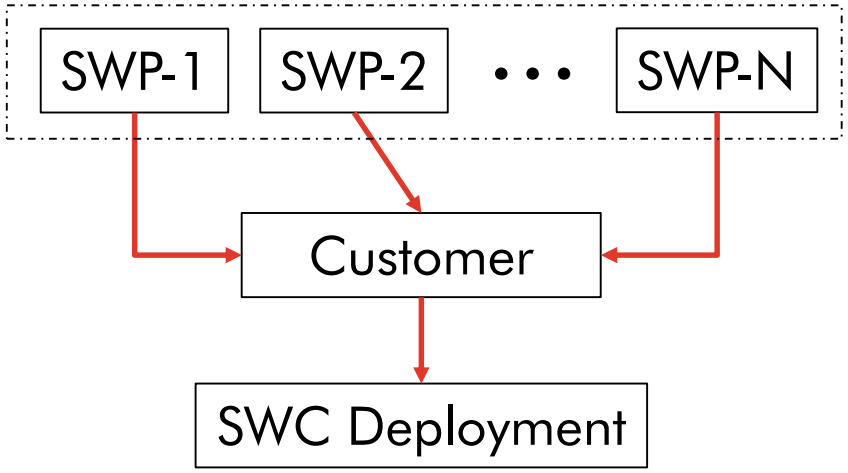
- N SW Providers (SWP)



Embedded SW Development

Vision

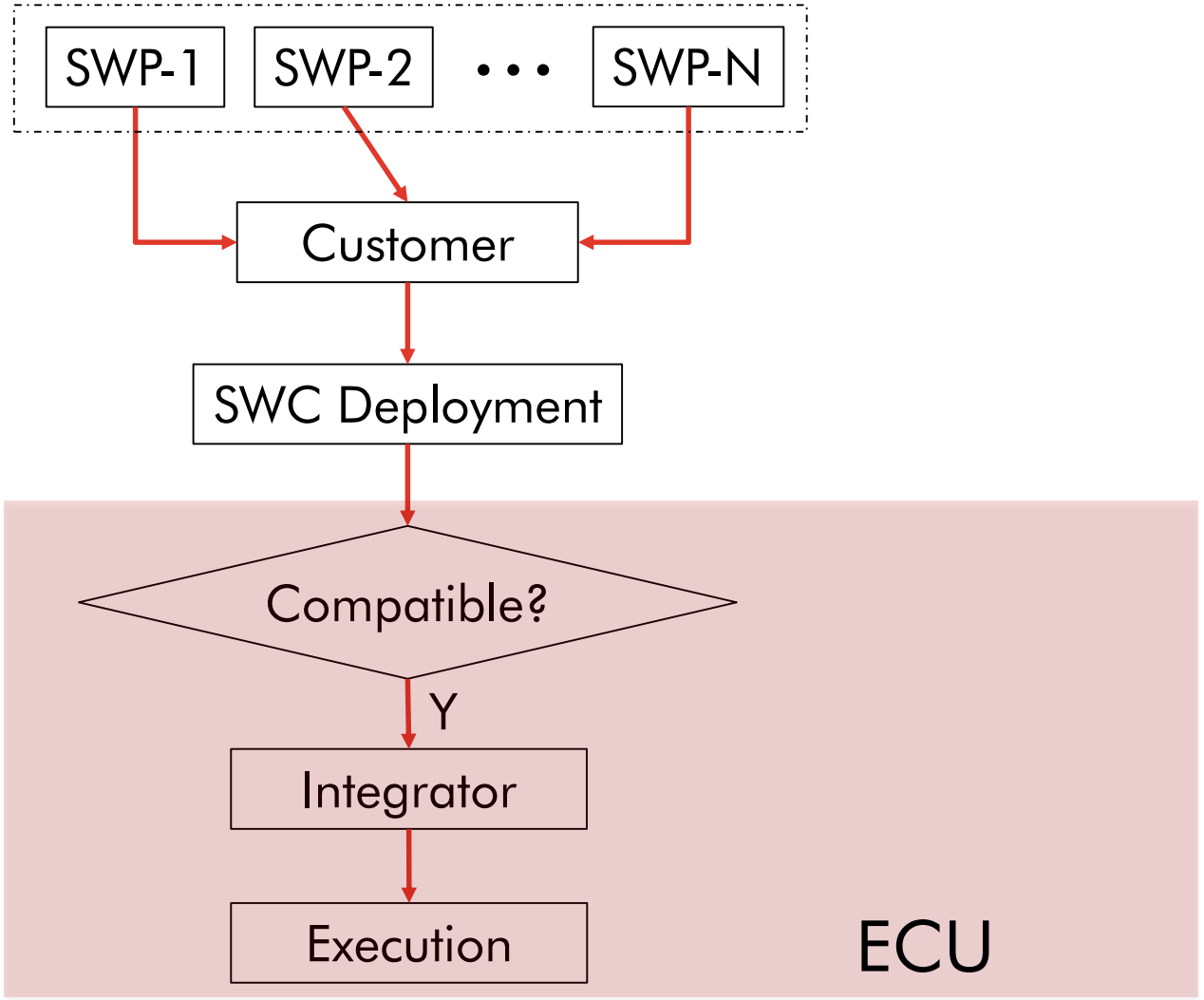
- N SW Providers (SWP)
- Software Components (SWCs) are deployed



Embedded SW Development

Vision

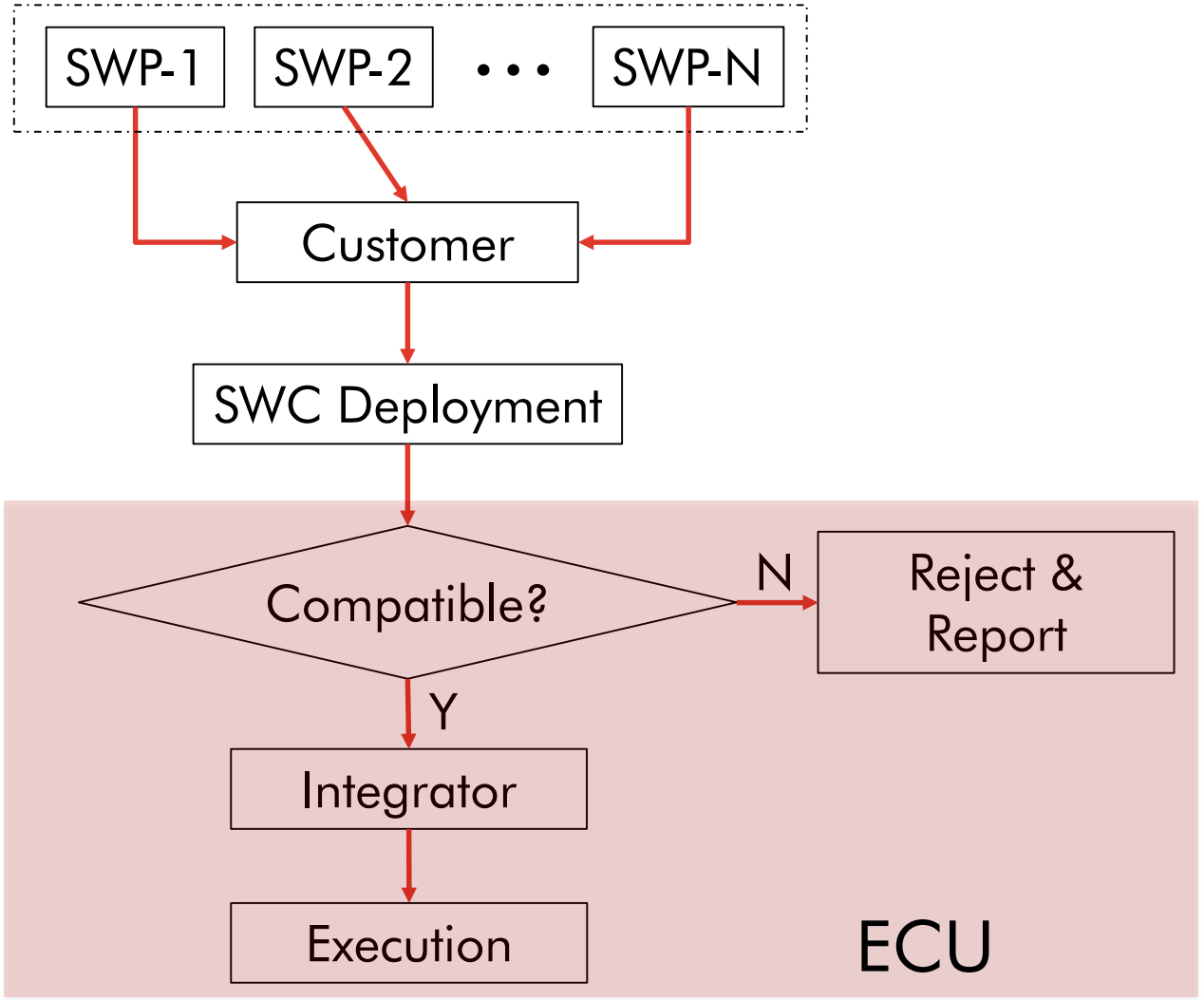
- N SW Providers (SWP)
- Software Components (SWCs) are deployed
- **Devices are SW Integrators**
  - Automatic



Embedded SW Development

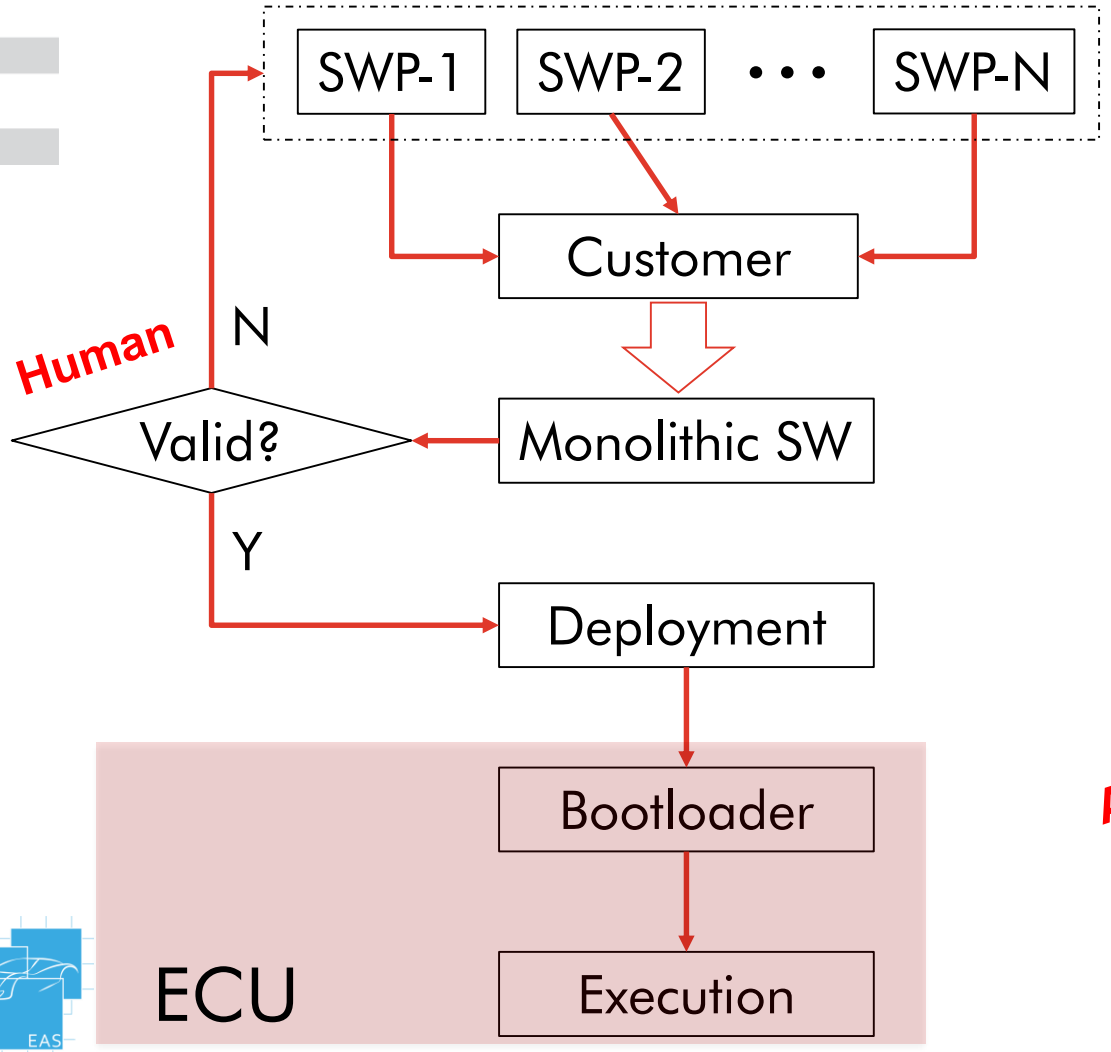
Vision

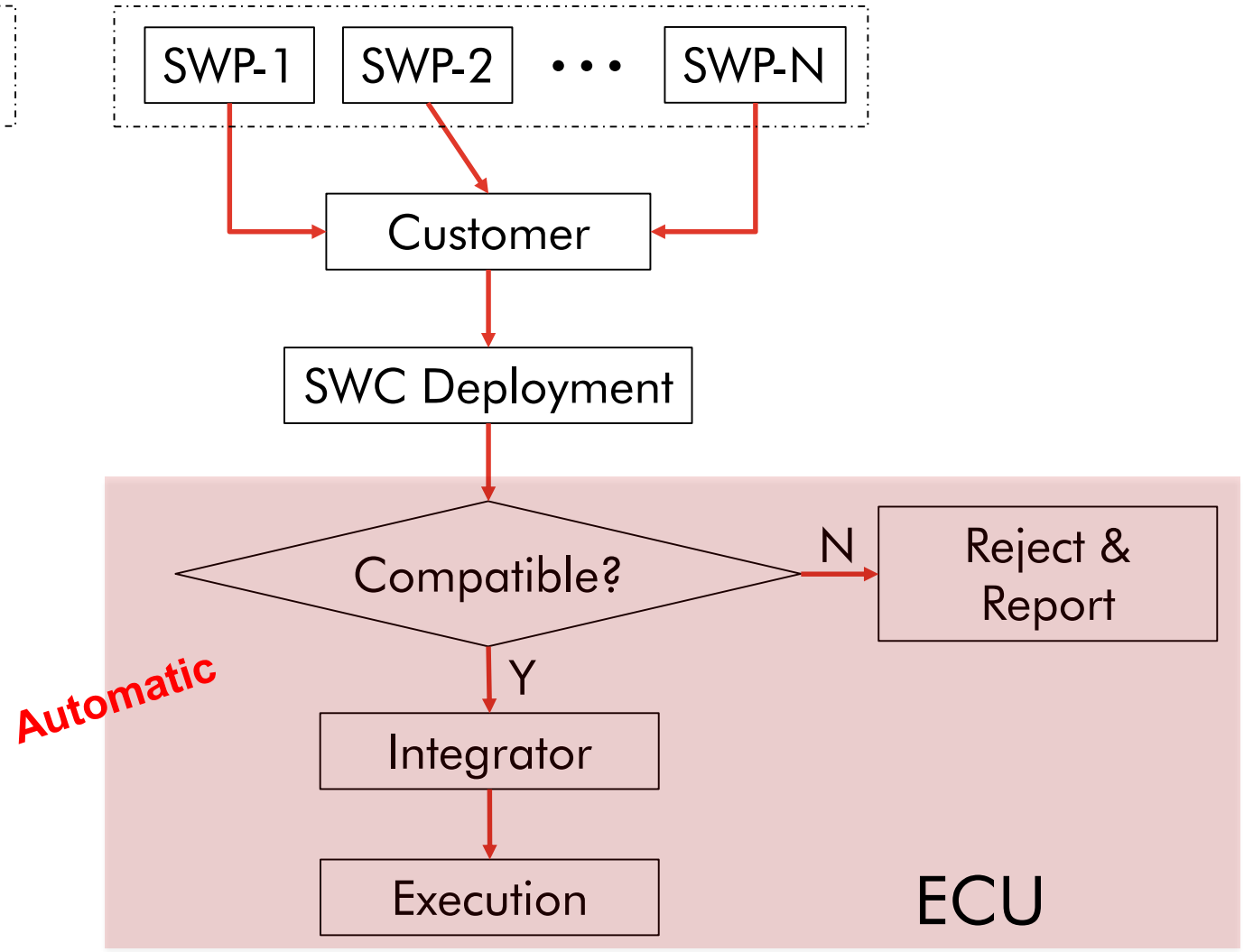
- N SW Providers (SWP)
- Software Components (SWCs) are deployed
- **Devices are SW Integrators**
  - Automatic



Embedded SW Development

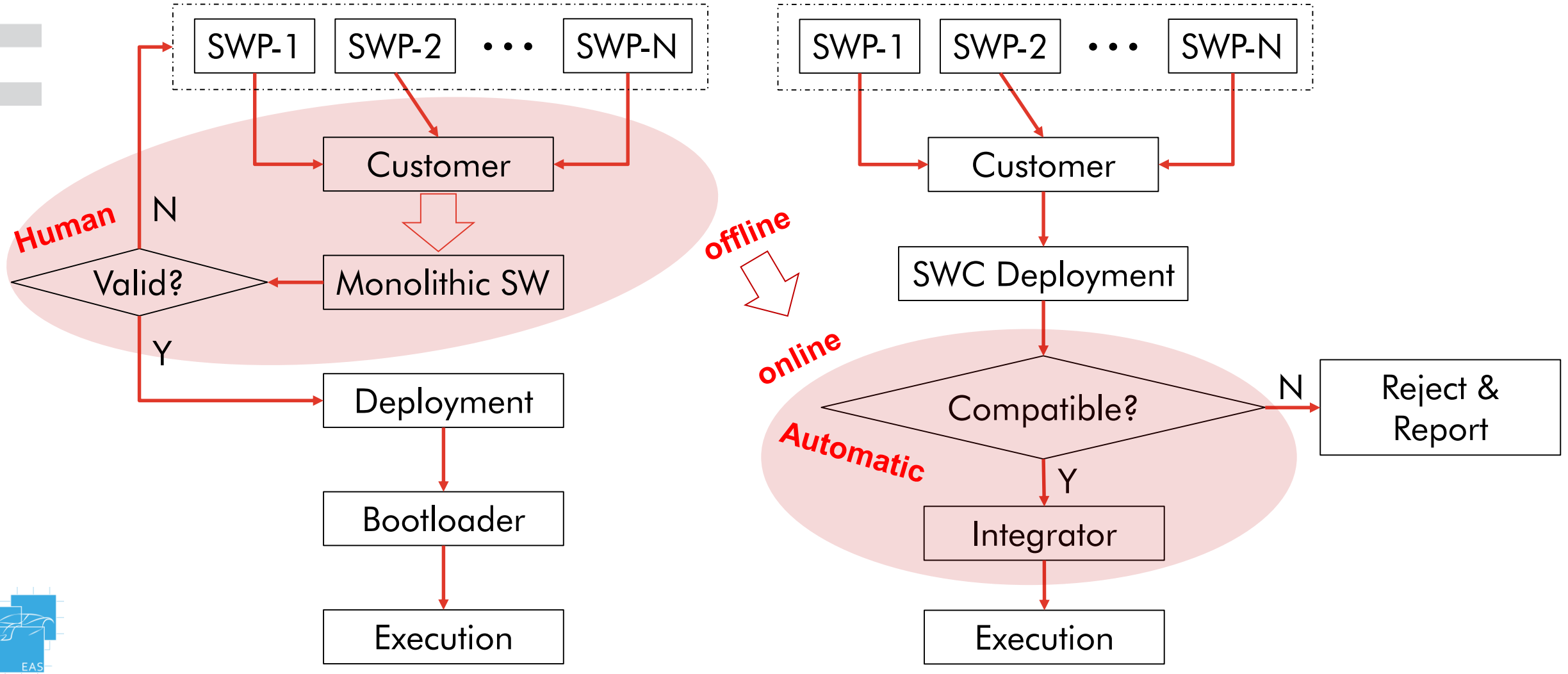
State of the Art / Vision





Embedded SW Development

State of the Art / Vision



Embedded SW Development

Research Questions

I. Can human-supervised integration be formalized/automated?

Embedded SW Development

Research Questions

I. Can human-supervised integration be formalized/automated?

II. What (additional) information is required for automatic integration?

Embedded SW Development

Research Questions

I. Can human-supervised integration be formalized/automated?

II. What (additional) information is required for automatic integration?

III. How can the information be obtained from modules?

Embedded SW Development

Research Questions

I. Can human-supervised integration be formalized/automated?

II. What (additional) information is required for automatic integration?

III. How can the information be obtained from modules?

IV. How must the developer add information to code/models/etc?

Embedded SW Development

Research Questions

I. Can human-supervised integration be formalized/automated?

II. What (additional) information is required for automatic integration?

III. How can the information be obtained from modules?

IV. How must the developer add information to code/models/etc?

V. How can the information be automatically combined across modules to verify system properties?

Embedded SW Development

Research Questions

I. Can human-supervised integration be formalized/automated?

II. What (additional) information is required for automatic integration?

III. How can the information be obtained from modules?

IV. How must the developer add information to code/models/etc?

V. How can the information be automatically combined across modules to verify system properties?

VI. Can automatic integration be performed on resource-constrained devices?

Embedded SW Development

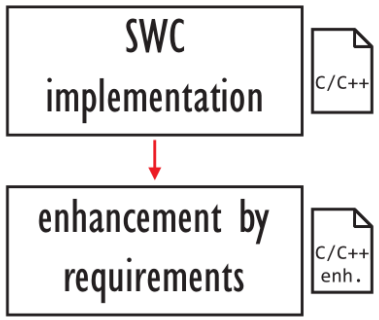
Vision – A closer Look

Module-Contained Development:

Embedded SW Development

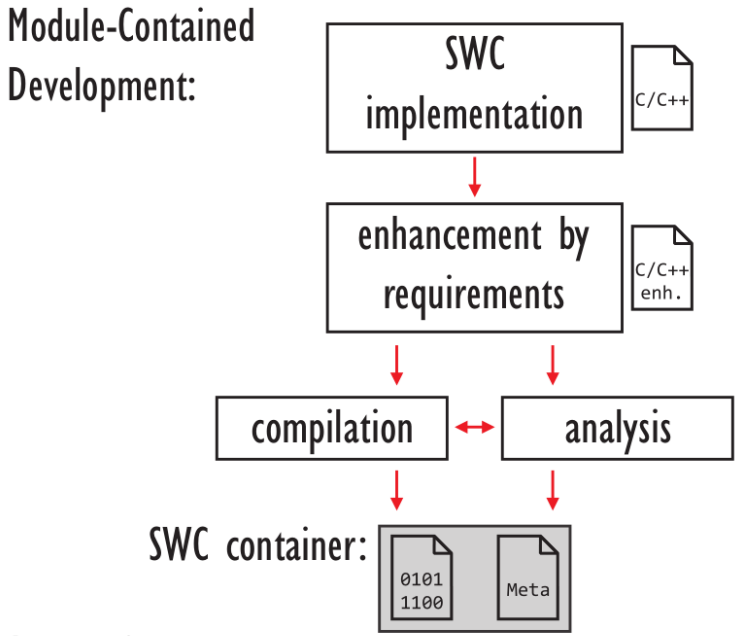
Vision – A closer Look

Module-Contained Development:



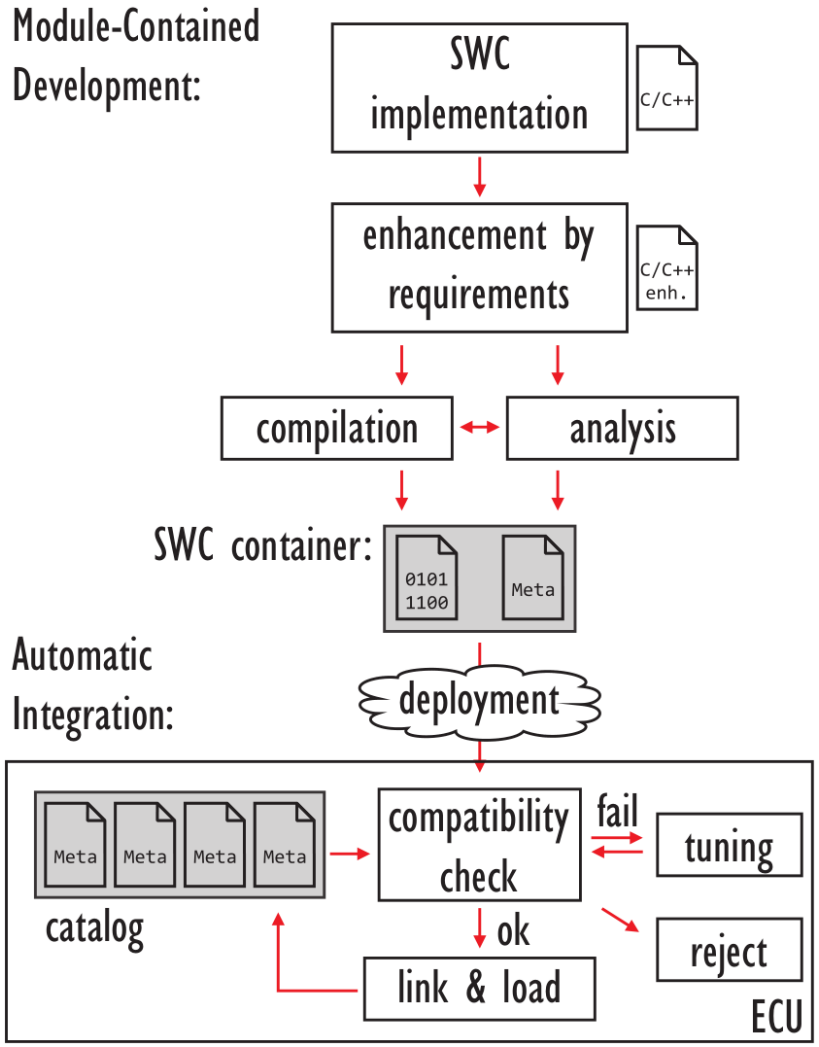
Embedded SW Development

Vision – A closer Look

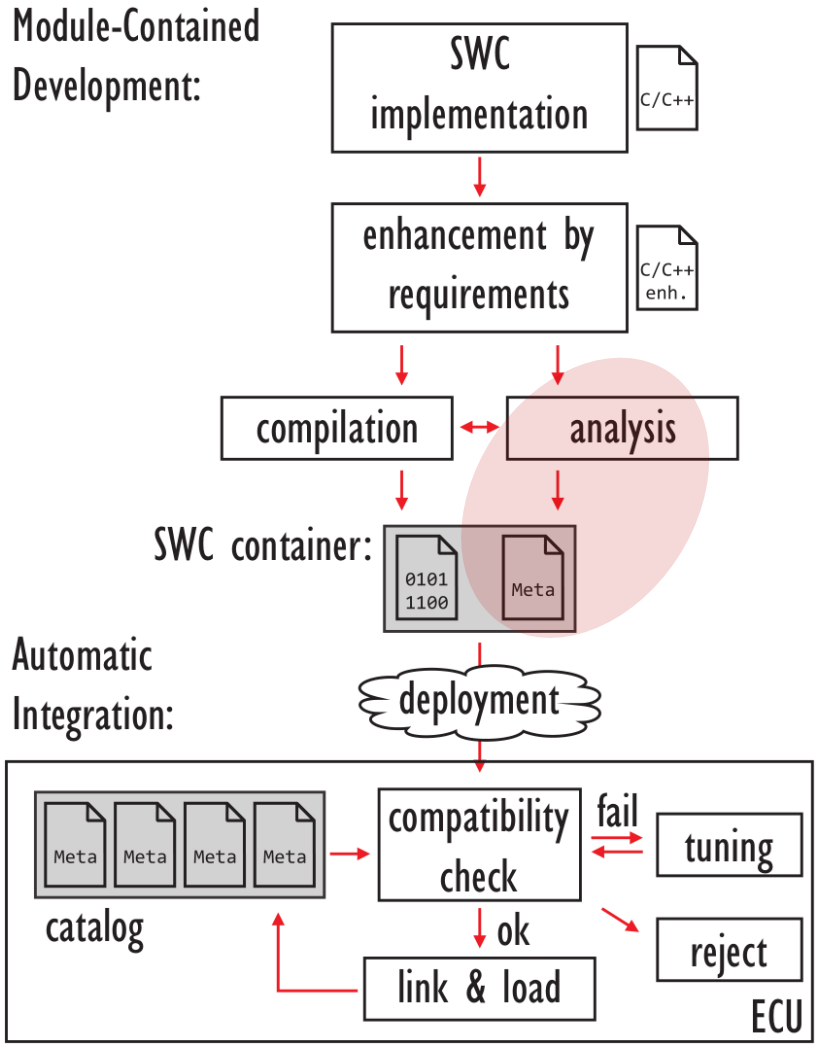


Embedded SW Development

Vision – A closer Look



Metadata Extraction



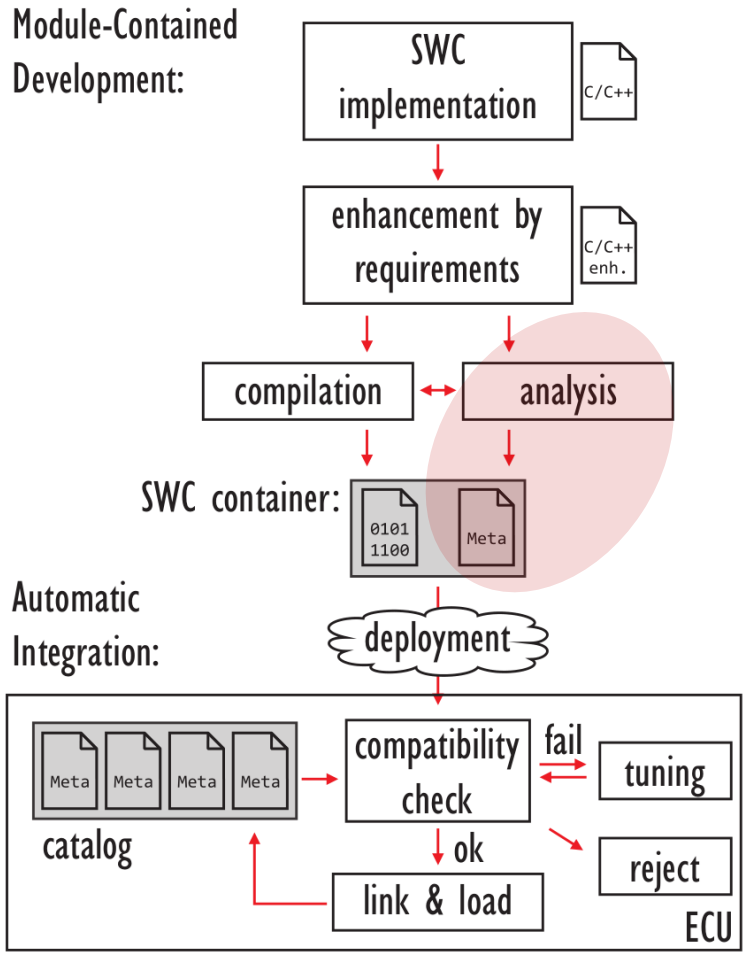
COFIE

COntrol Flow and Interaction Expression

L. B. Ribeiro and M. Baunach, "COFIE: a regex-like interaction and control flow description,"
2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), Taipei, Taiwan, 2019, pp. 67-72.

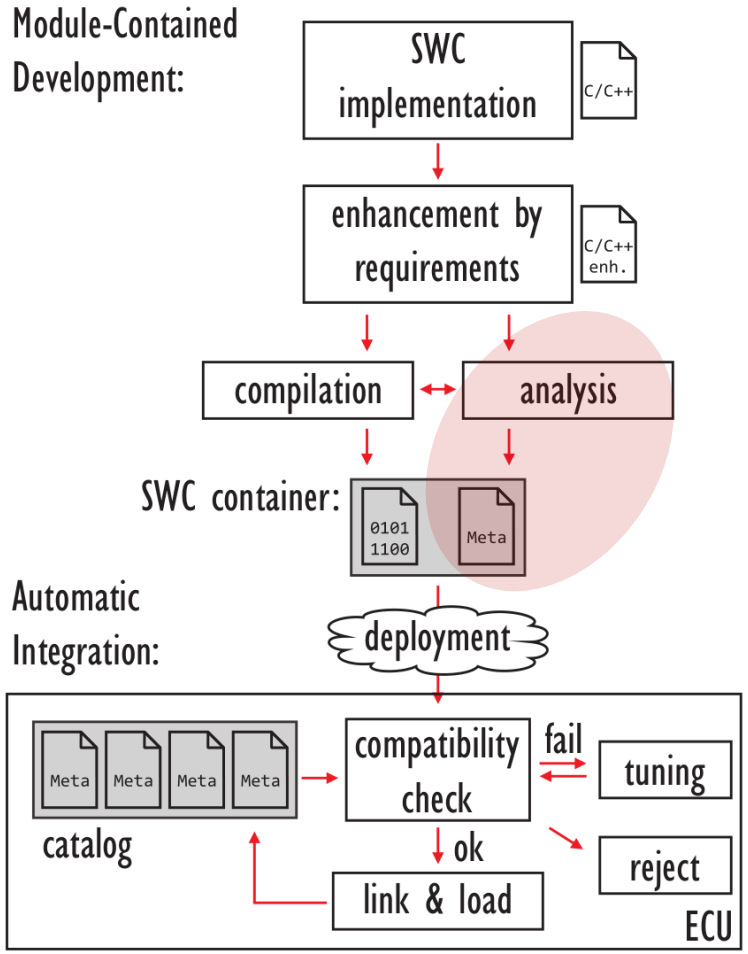
COFIE

- Metadata Extraction
  - Control Flow
  - Synchronization
  - Mutual Exclusion



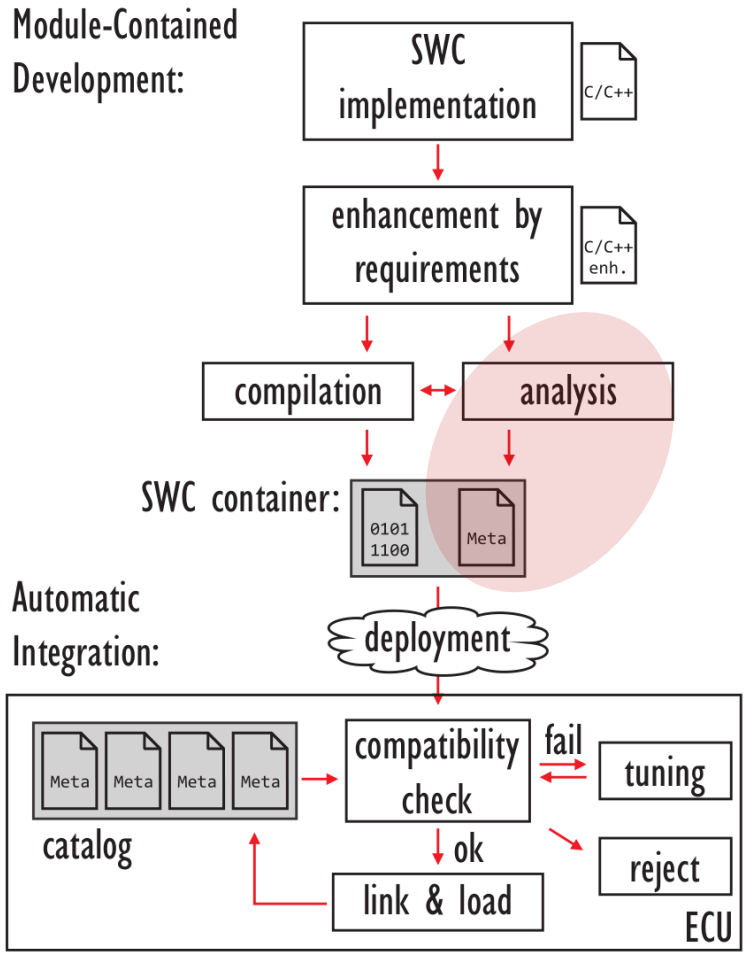
COFIE

- Metadata Extraction
  - Control Flow
  - Synchronization
  - Mutual Exclusion
- Regex-like Notation



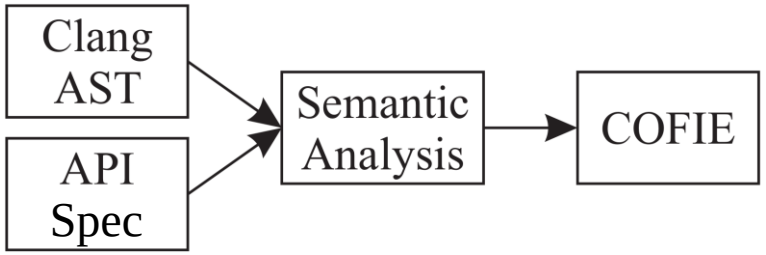
COFIE

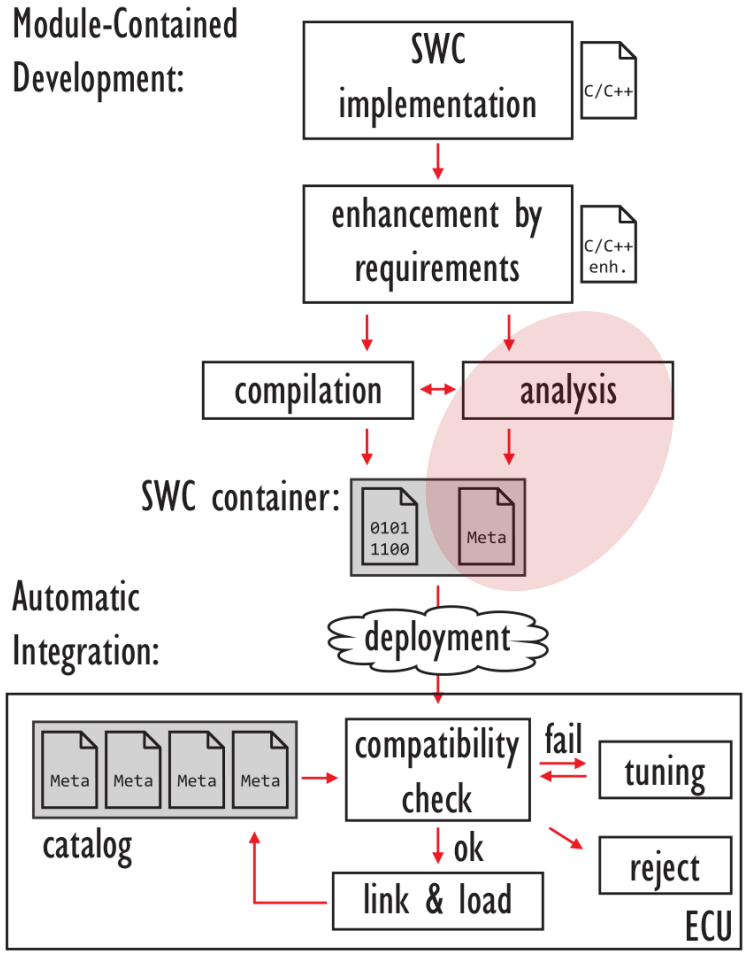
- Metadata Extraction
  - Control Flow
  - Synchronization
  - Mutual Exclusion
- Regex-like Notation
- No Exposure of Implementation Details



COFIE

- Metadata Extraction
  - Control Flow
  - Synchronization
  - Mutual Exclusion
- Regex-like Notation
- No Exposure of Implementation Details
- Transparent to SW Developer





COFIE

MCSmartOS Basic API

API	COFIE Term
`getResource(Resource_t r1)`	Gr1
`releaseResource(Resource_t r1)`	Rr1
`waitEvent(Event_t e1)`	We1
`setEvent(Event_t e1)`	Se1
`notifyEvent(Event_t e1)`	Ne1

COFIE

Examples - MCSmartOS Basic API

```
void funcC1 () {
  if (cond){
    getResource(r1);
    // critical code
  }
  releaseResource(r1);
}
```

COFIE: Gr1?Rr1

COFIE

Examples - MCSmartOS Basic API

```
void funcC1 () {
  if (cond){
    getResource(r1);
    // critical code
  }
  releaseResource(r1);
}
```

COFIE: Gr1?Rr1

```
void funcC2 () {
  if (cond) getResource(r1);
  else getResource(r2);
  // critical code
  releaseResource(r1);
}
```

COFIE: (Gr1|Gr2)Rr1

MCSmartOS API

MCSmartOS Time-Aware API and Examples

API	COFIE Term
`getResourceFor(Resource_t r, Time_t t)`	Gtr
`waitEventFor(Event_t e, Time_t t)`	Wte

```
void funcT () {
  waitEventFor(e, 200);
  // some code
}
```

COFIE: W200e

MCSmartOS API

MCSmartOS Time-Aware API and Examples

API	COFIE Term
`getResourceFor(Resource_t r, Time_t t)`	Gtr
`waitEventFor(Event_t e, Time_t t)`	Wte

```
void funcT () {
  waitEventFor(e, 200);
  // some code
}
```

COFIE: W200e

```
void funcAPIC () {
  if (getResourceFor(r1,200)){
    // critical code
    releaseResource(r1);
  }
  waitEvent(e1);
}
```

COFIE: G200r1<Rr1>We1

COFIE

Neutral Code and Resource Leakage

```
TASK() {
  while (1) {                // (
    openDev();
    // ...
    getResource(r3);
    // ...
    releaseResource(r3);
    closeDev();
  }
}
```

COFIE

Neutral Code and Resource Leakage

```
TASK() {
  while (1) {                    // (
    openDev();
    // ...
    getResource(r3);
    // ...
    releaseResource(r3);
    closeDev();
  }
}
```

```
void openDev() {
  if (condition) {               // (
    getResource(r2);             // Gr2
    // ...
    releaseResource(r2);         // Rr2
  }                              // )?
  getResource(r1);               // Gr1
  // ...
}
```

Full COFIE: **(Gr2 Rr2)? Gr1**

Reduced COFIE: **Gr1 → not neutral**

COFIE

Neutral Code and Resource Leakage

```
TASK() {
  while (1) {                    // (
    openDev();                   // (Gr2 Rr2)? Gr1
    // ...
    getResource(r3);
    // ...
    releaseResource(r3);
    closeDev();
  }
}
```

```
void openDev() {
  if (condition) {               // (
    getResource(r2);             // Gr2
    // ...
    releaseResource(r2);         // Rr2
  }                              // )?
  getResource(r1);               // Gr1
  // ...
}
```

Full COFIE: **(Gr2 Rr2)? Gr1**

Reduced COFIE: **Gr1 → not neutral**

COFIE

Neutral Code and Resource Leakage

```
TASK() {
  while (1) {                 // (
    openDev();                // (Gr2 Rr2)? Gr1
    // ...
    getResource(r3);          // Gr3
    // ...
    releaseResource(r3);      // Rr3
    closeDev();
  }
}
```

```
void openDev() {
  if (condition) {            // (
    getResource(r2);          // Gr2
    // ...
    releaseResource(r2);      // Rr2
  }                           // )?
  getResource(r1);            // Gr1
  // ...
}
```

Full COFIE: (Gr2 Rr2)? Gr1
Reduced COFIE: Gr1 → not neutral

COFIE

Neutral Code and Resource Leakage

```
TASK() {
  while (1) {                 // (
    openDev();                // (Gr2 Rr2)? Gr1
    // ...
    getResource(r3);          // Gr3
    // ...
    releaseResource(r3);      // Rr3
    closeDev();
  }
}
```

```
void openDev() {
  if (condition) {            // (
    getResource(r2);          // Gr2
    // ...
    releaseResource(r2);      // Rr2
  }                           // )?
  getResource(r1);            // Gr1
  // ...
}

Full COFIE:      (Gr2 Rr2)? Gr1
Reduced COFIE:   Gr1 → not neutral
```

```
void closeDev() {
  releaseResource(r1);        // Rr1
  // ...
}

Full COFIE:      Rr1
Reduced COFIE:   Rr1 → not neutral
```

COFIE

Neutral Code and Resource Leakage

```
TASK() {
  while (1) {                 // (
    openDev();                // (Gr2 Rr2)? Gr1
    // ...
    getResource(r3);          // Gr3
    // ...
    releaseResource(r3);      // Rr3
    closeDev();               // Rr1
  }                           // )+
}
```

```
void openDev() {
  if (condition) {            // (
    getResource(r2);          // Gr2
    // ...
    releaseResource(r2);      // Rr2
  }                           // )?
  getResource(r1);            // Gr1
  // ...
}

Full COFIE:      (Gr2 Rr2)? Gr1
Reduced COFIE:   Gr1 → not neutral
```

```
void closeDev() {
  releaseResource(r1);        // Rr1
  // ...
}

Full COFIE:      Rr1
Reduced COFIE:   Rr1 → not neutral
```

COFIE

Neutral Code and Resource Leakage

```
TASK() {
  while (1) {                 // (
    openDev();                // (Gr2 Rr2)? Gr1
    // ...
    getResource(r3);          // Gr3
    // ...
    releaseResource(r3);      // Rr3
    closeDev();               // Rr1
  }                           // )+
}
```

Full COFIE: ((Gr2 Rr2)? Gr1 Gr3 Rr3 Rr1)+
Reduced COFIE: Empty → neutral
→ no resource leakage

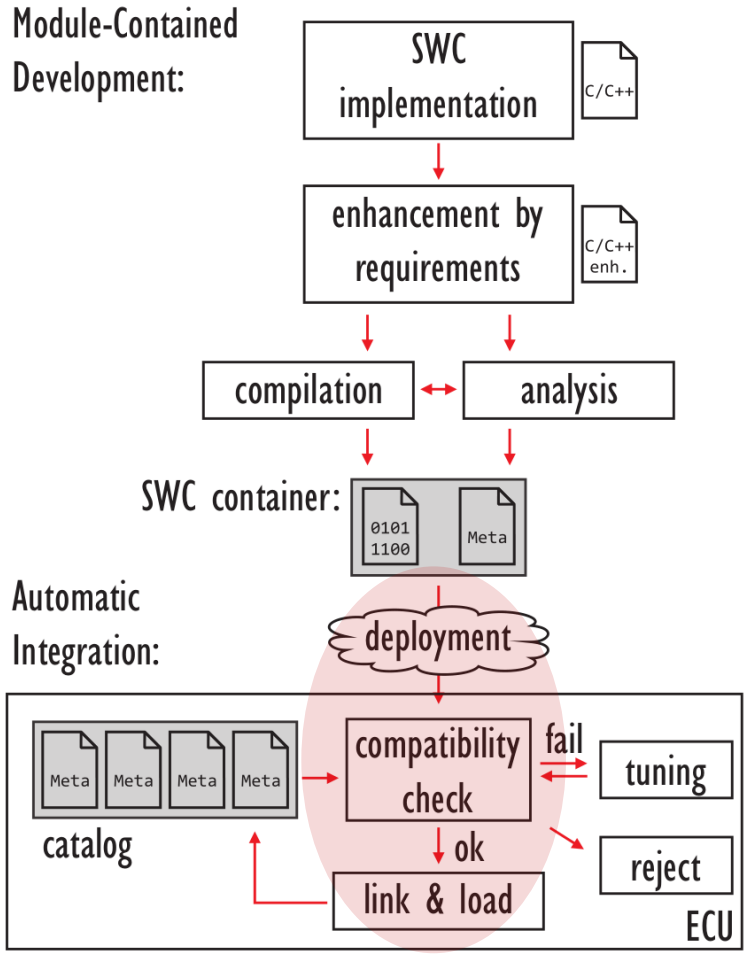
```
void openDev() {
  if (condition) {            // (
    getResource(r2);          // Gr2
    // ...
    releaseResource(r2);      // Rr2
  }                           // )?
  getResource(r1);            // Gr1
  // ...
}
```

Full COFIE: (Gr2 Rr2)? Gr1
Reduced COFIE: Gr1 → not neutral

```
void closeDev() {
  releaseResource(r1);        // Rr1
  // ...
}
```

Full COFIE: Rr1
Reduced COFIE: Rr1 → not neutral

Modular Updates and Update Protocol

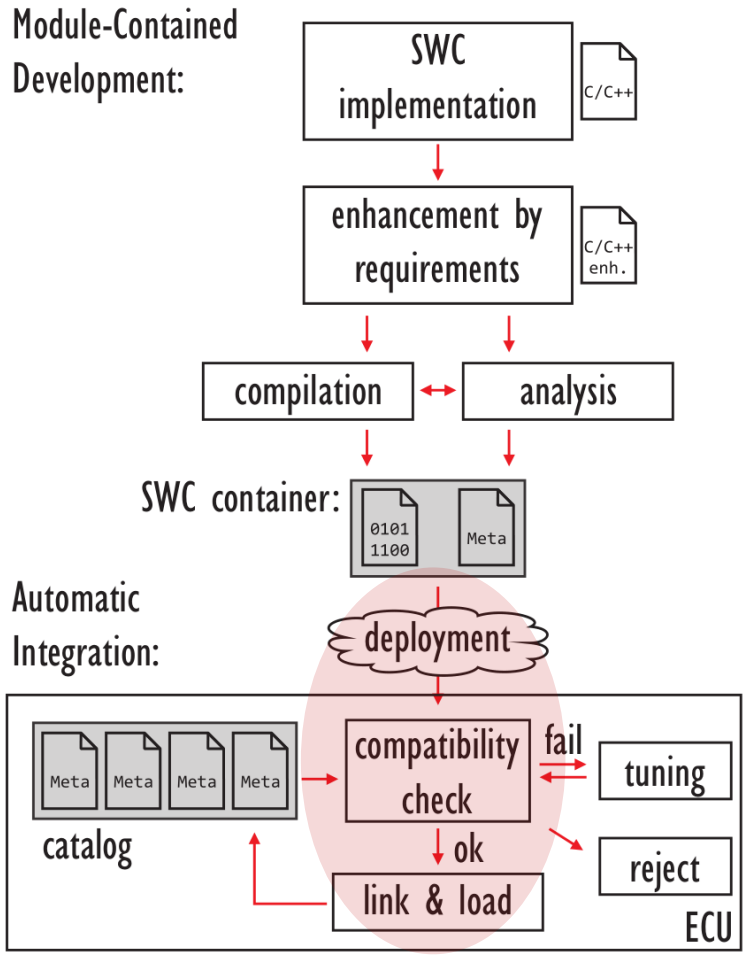


** L. B. Ribeiro and M. Baunach,
"Towards Automatic SW Integration in Dependable Embedded Systems,"
International Conference on Embedded Wireless Systems and Networks (EWSN) 2020.
** Under Review

Modular Updates

- Rebootless

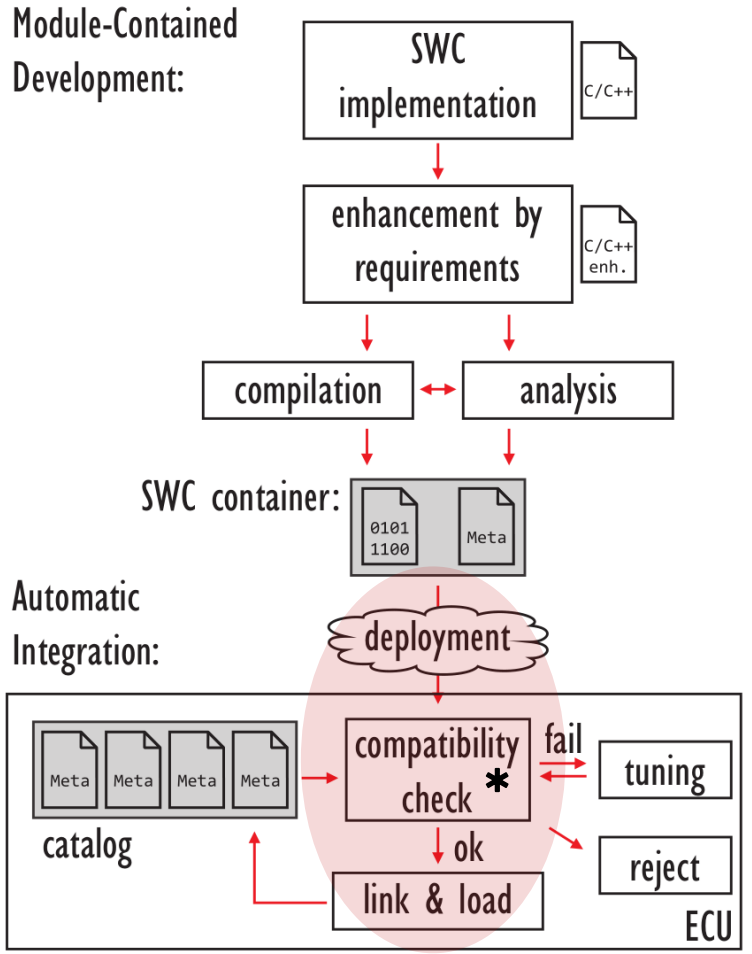
MCSmart OS
Multi-Core Sustainable modular adept real-time Operating System



Modular Updates

MCSmartOS
Multi-Core Sustainable modular adept real-time Operating System

- Rebootless
- Compatibility Check Operations
  - Pluggability Check
  - Interoperability Check*



* Ongoing

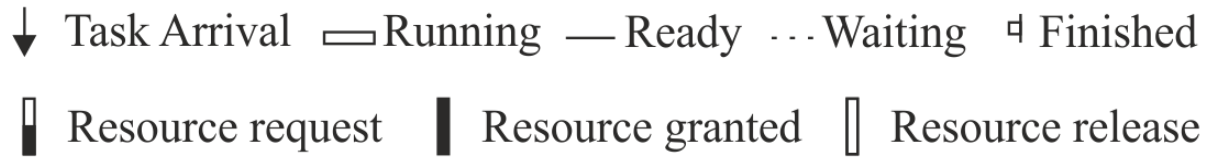
Modular Updates

Interoperability Check - COFIE

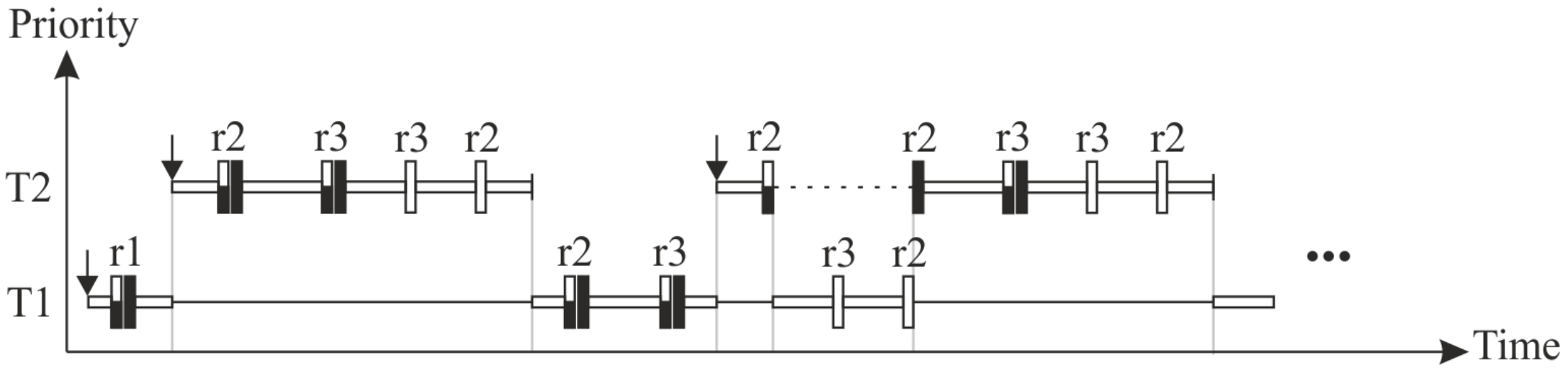
Task	COFIE	Priority
T1	Gr1Gr2Rr2Rr1	Medium
T2	Gr2Gr3Rr3Rr2	High

Modular Updates

Interoperability Check - COFIE

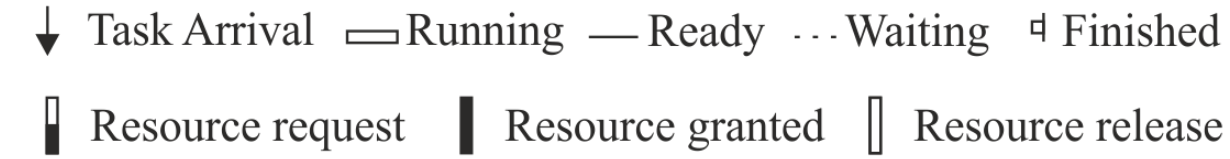


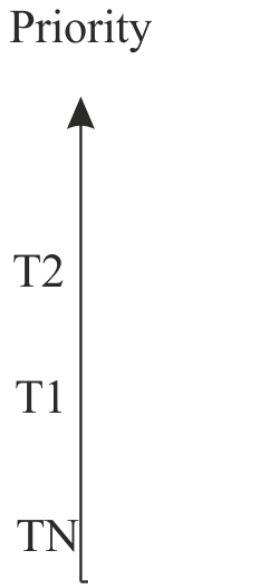
Task	COFIE	Priority
T1	Gr1Gr2Rr2Rr1	Medium
T2	Gr2Gr3Rr3Rr2	High



Modular Updates

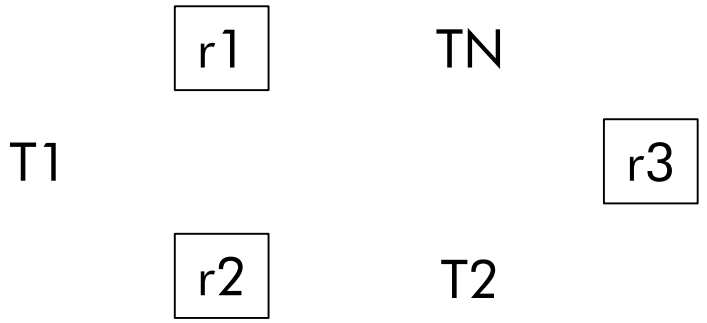
Interoperability Check - COFIE





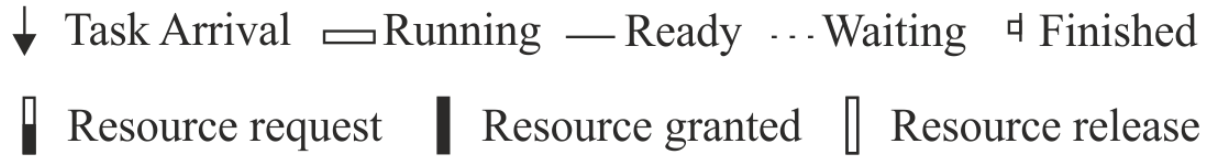
Task	COFIE	Priority
TN (to be added)	Gr3Gr1Rr1Rr3	Low
T1 (in system)	Gr1Gr2Rr2Rr1	Medium
T2 (in system)	Gr2Gr3Rr3Rr2	High

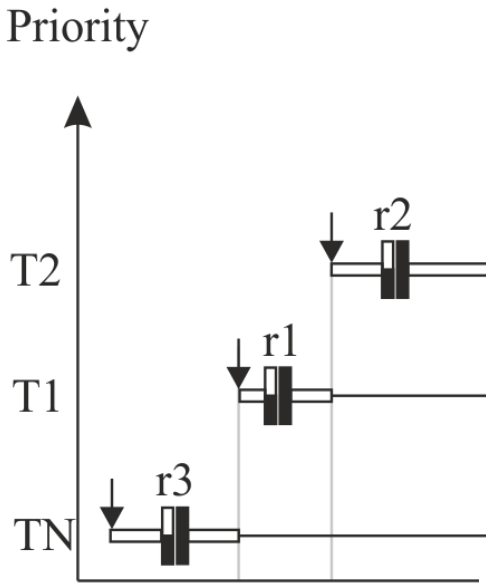
Resource Allocation Graph



Modular Updates

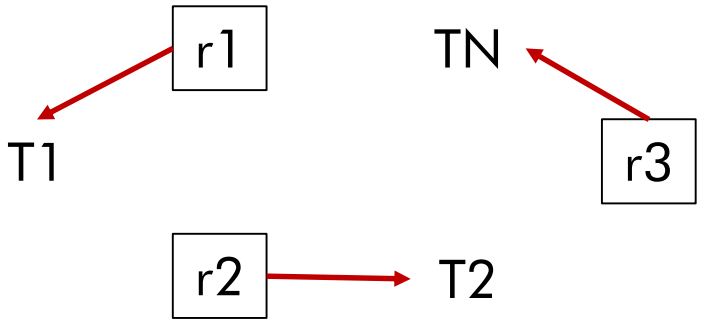
Interoperability Check - COFIE





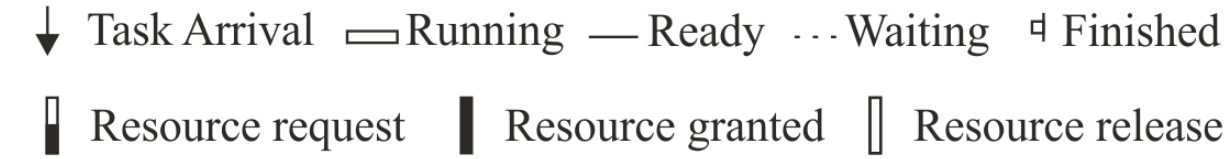
Task	COFIE	Priority
TN (to be added)	Gr3Gr1Rr1Rr3	Low
T1 (in system)	Gr1Gr2Rr2Rr1	Medium
T2 (in system)	Gr2Gr3Rr3Rr2	High

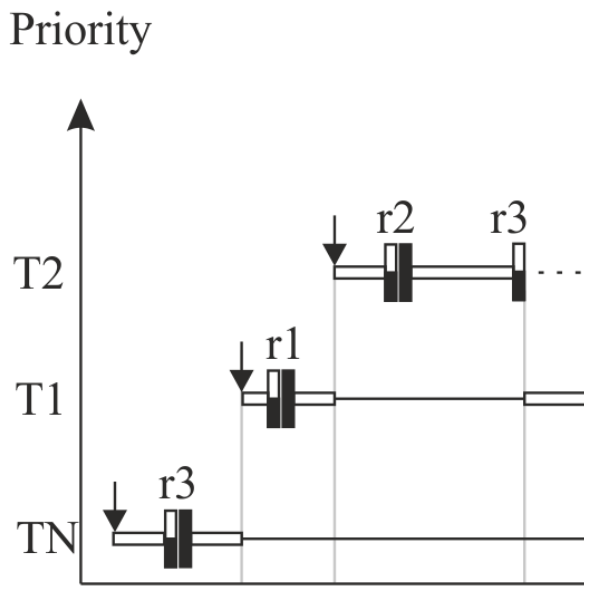
Resource Allocation Graph



Modular Updates

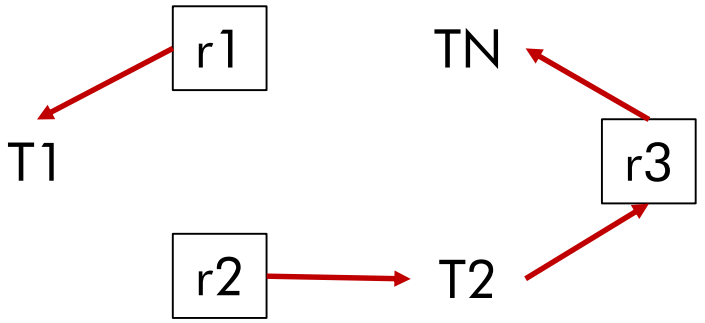
Interoperability Check - COFIE





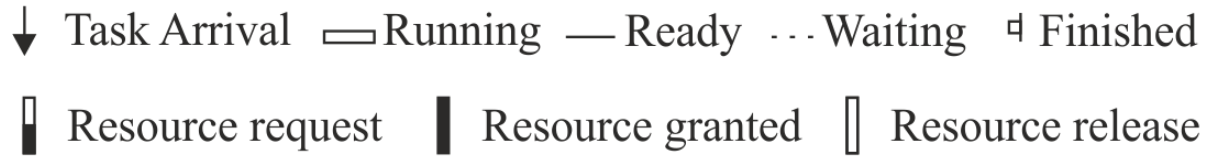
Task	COFIE	Priority
TN (to be added)	Gr3Gr1Rr1Rr3	Low
T1 (in system)	Gr1Gr2Rr2Rr1	Medium
T2 (in system)	Gr2Gr3Rr3Rr2	High

Resource Allocation Graph

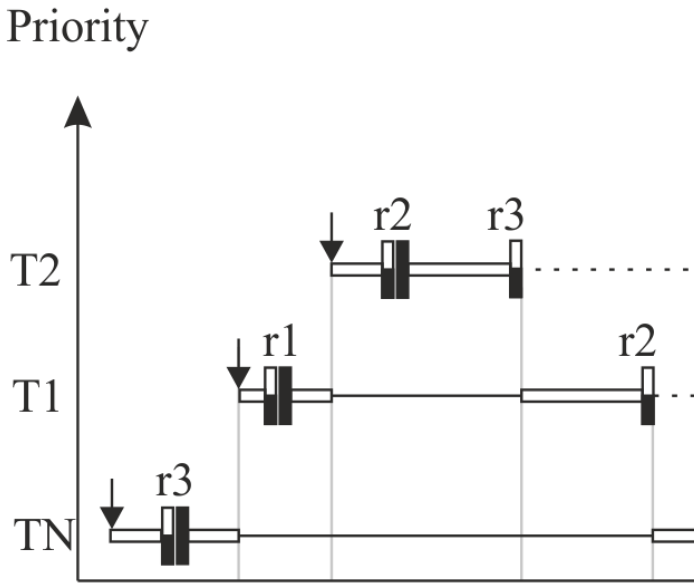


Modular Updates

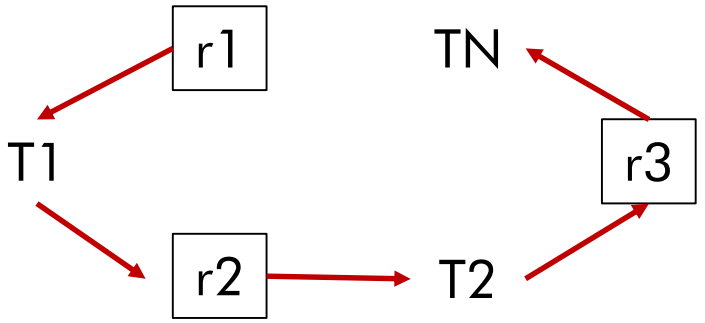
Interoperability Check - COFIE



Task	COFIE	Priority
TN (to be added)	Gr3Gr1Rr1Rr3	Low
T1 (in system)	Gr1Gr2Rr2Rr1	Medium
T2 (in system)	Gr2Gr3Rr3Rr2	High

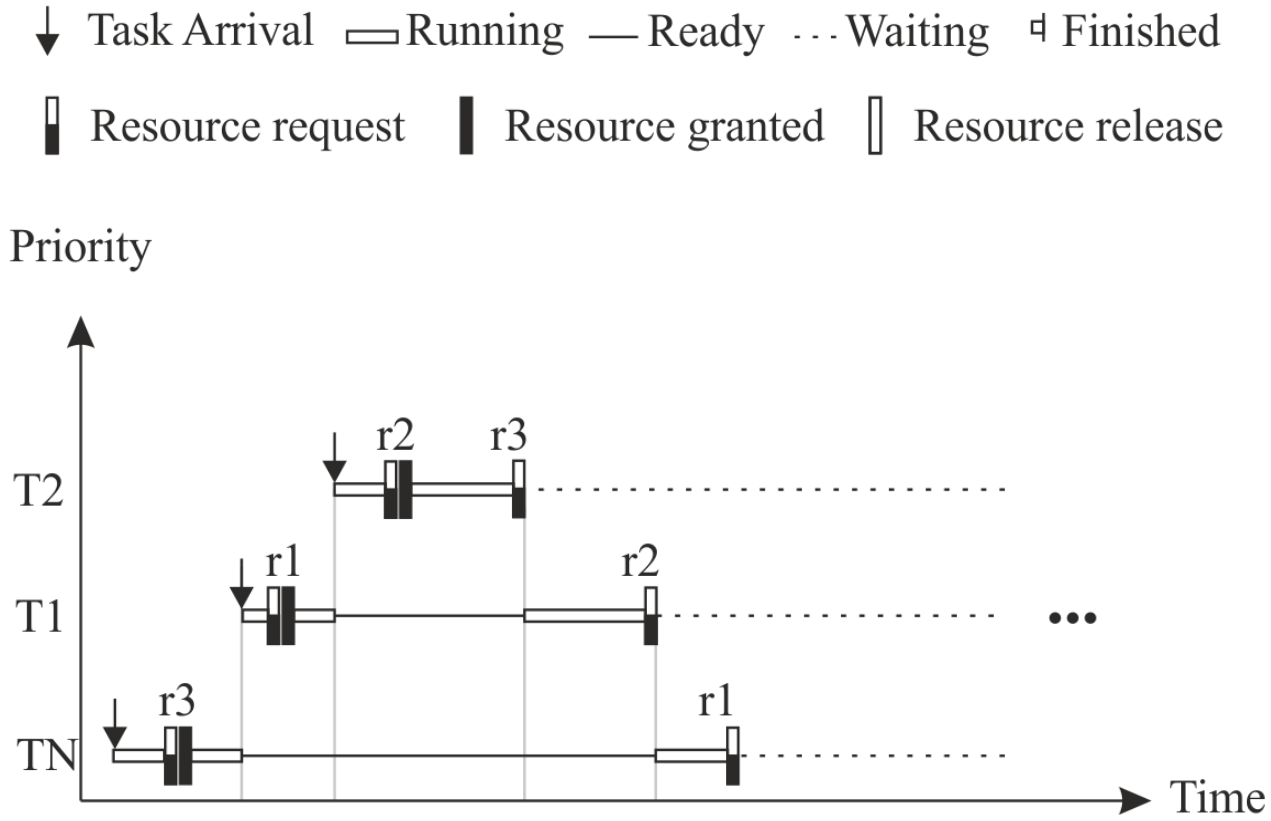


Resource Allocation Graph



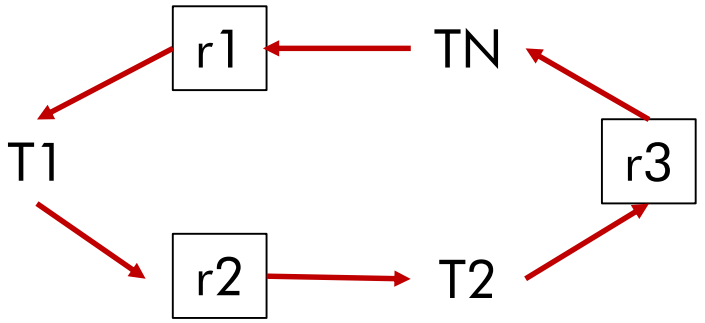
Modular Updates

Interoperability Check - COFIE



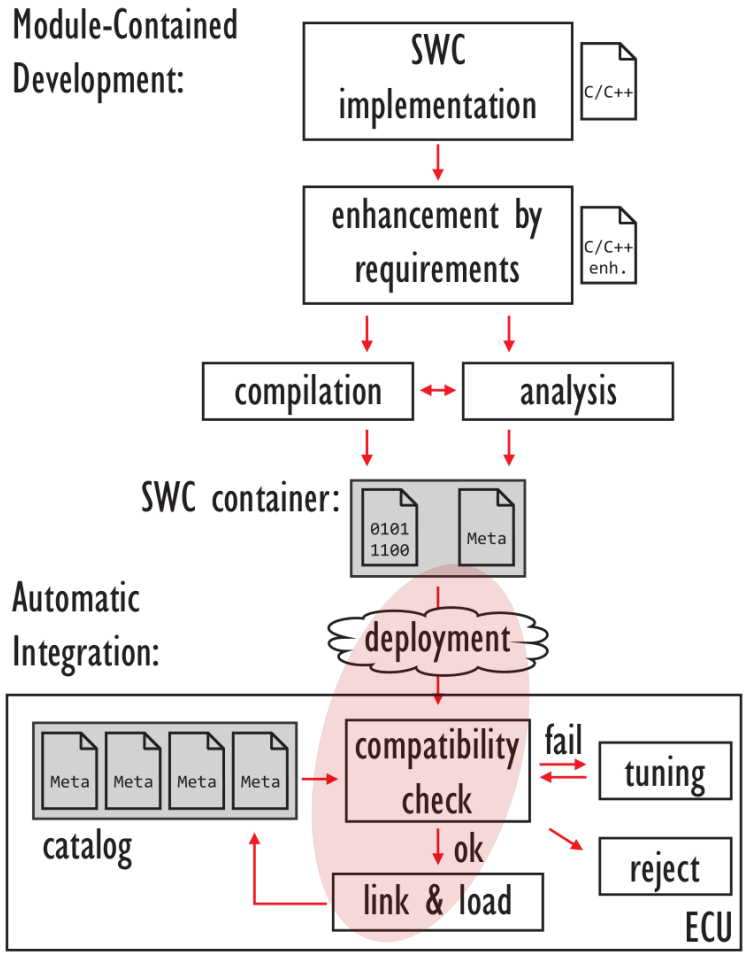
Task	COFIE	Priority
TN (to be added)	Gr3Gr1Rr1Rr3	Low
T1 (in system)	Gr1Gr2Rr2Rr1	Medium
T2 (in system)	Gr2Gr3Rr3Rr2	High

Resource Allocation Graph



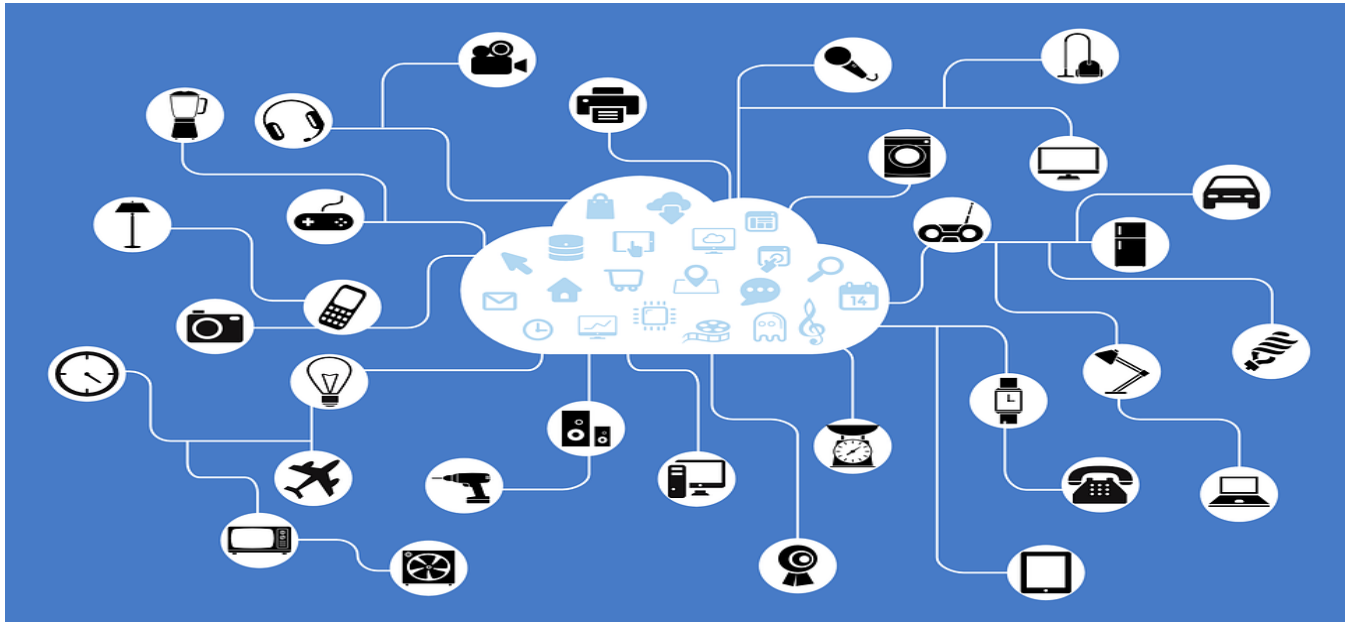
Update Protocol

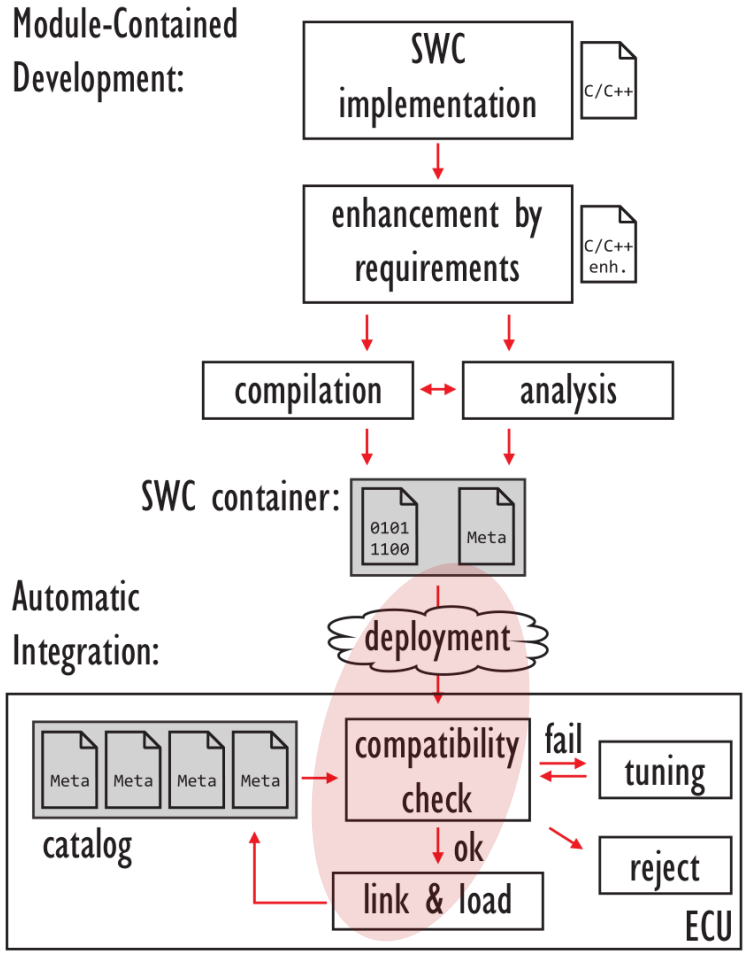
- Deployment



Update Protocol

- Deployment
  - Unified Update Protocol
    - Cope with HW Diversity
    - Compatibility Check Distribution





Update Protocol

Operations Performed by Different Devices

Performance	Pluggability	Relocation	Linking	Interoperability
Lowest				
Low	x			
Medium	x	x		
High	x	x	x	
Highest	x	x	x	x

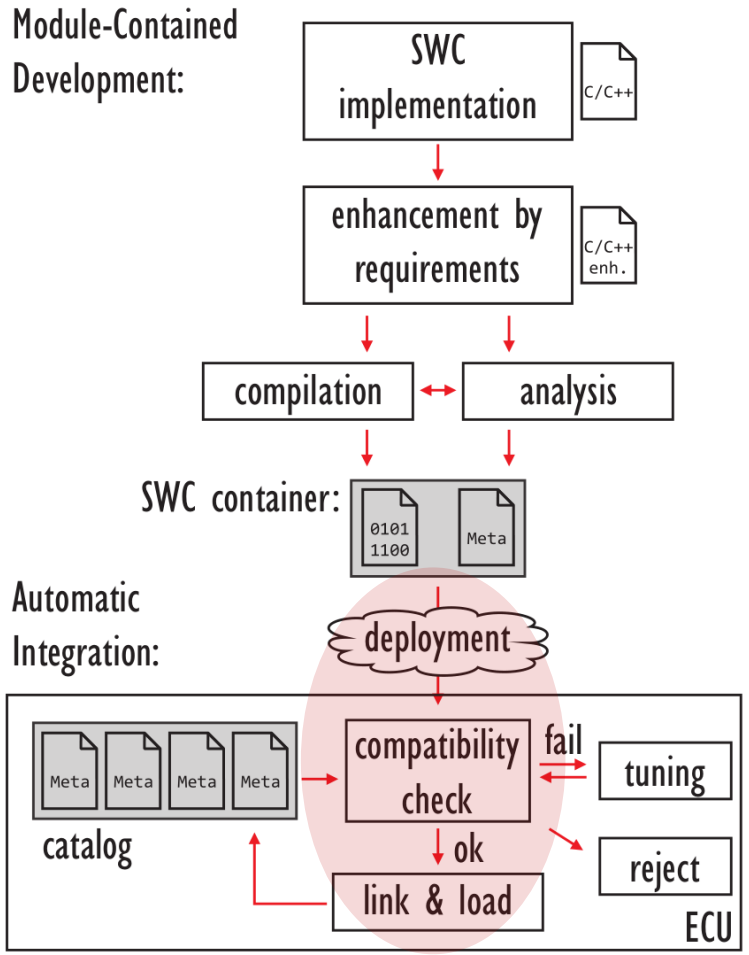
Update Protocol

Operations Performed on Embedded Devices

Performance	Pluggability	Relocation	Linking	Interoperability
Lowest				
Low	x			
Medium	x	x		
High	x	x	x	
Highest	x	x	x	x

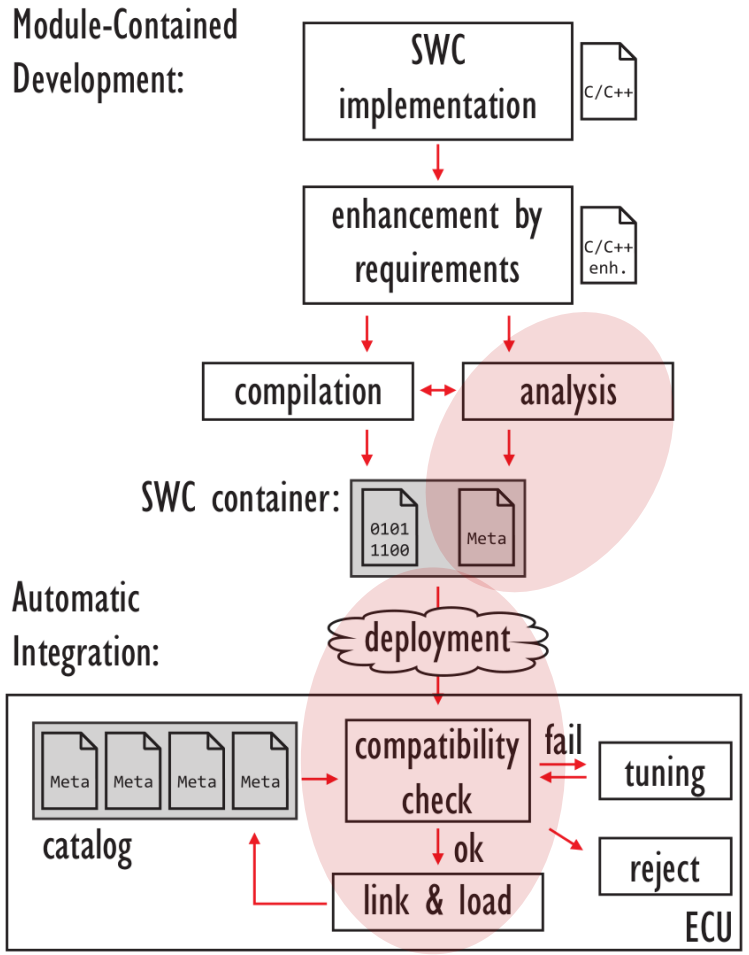
Summary and Future Work

- Partial Updates
  - OS Support and Protocol



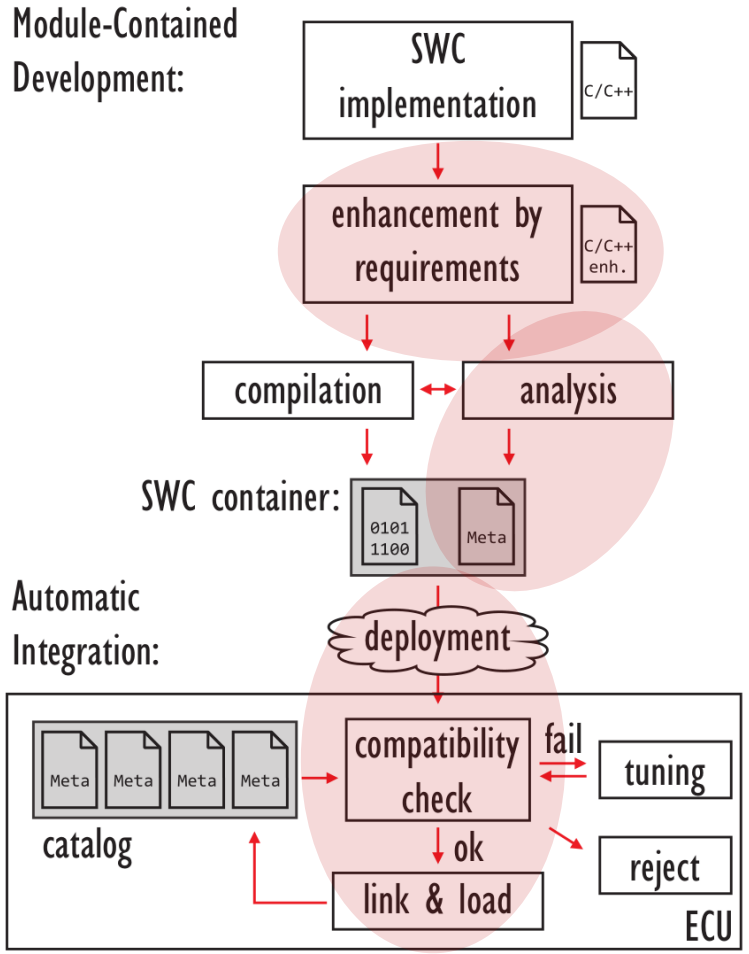
Summary and Future Work

- Partial Updates
  - OS Support and Protocol
- Metadata Generation
  - COFIE
    - Control Flow and Tasks Interaction



Summary and Future Work

- Partial Updates
  - OS Support and Protocol
- Metadata Generation ⟺ Interoperability Check
  - COFIE
    - Control Flow and Tasks Interaction
  - WCET/WCRT
  - Data-Flow
  - Non-Functional Requirements/Properties
  - …



Thank you!

Leandro Batista Ribeiro
lbatistaribeiro@tugraz.at

Marcel Baunach
baunach@tugraz.at

Institute of Technical Informatics
Embedded Automotive Systems Group
Graz University of Technology