SmartOS: An OS Architecture for Sustainable Embedded Systems

Tobias Scheipel
tobias.scheipel@tugraz.at

Leandro Batista Ribeiro
lbatistaribeiro@tugraz.at

Tim Sagaster
tim.sagaster@student.tugraz.at

Marcel Baunach
baunach@tugraz.at

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

Outline

SmartOS
Sustainable modular adaptive real-time Operating System

- Introduction and Motivation
- OS Architecture and Basic Concepts
- Extended Concepts and Special Features
  - MCU/OS Co-Design
  - Compositional Software and Automatic Integration
  - Formal Methods for Verification and Portability
- Conclusion

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Demands:

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Demands:

D1 Partial Software Updates

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Demands:

D1 Partial Software Updates

D2 Support for Hardware Modification

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Demands:

D1 Partial Software Updates

D2 Support for Hardware Modification

D3 Automatic Integration

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Demands:

D1 Partial Software Updates

D2 Support for Hardware Modification

D3 Automatic Integration

D4 Hard Correctness Guarantees for Composition

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Demands:

D1 Partial Software Updates
D2 Support for Hardware Modification
D3 Automatic Integration
D4 Hard Correctness Guarantees for Composition
D5 Efficient and Automatic Portability

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Demands:

D1 Partial Software Updates
D2 Support for Hardware Modification
D3 Automatic Integration
D4 Hard Correctness Guarantees for Composition
D5 Efficient and Automatic Portability

Possibilities:

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Demands:

D1 Partial Software Updates
D2 Support for Hardware Modification
D3 Automatic Integration
D4 Hard Correctness Guarantees for Composition
D5 Efficient and Automatic Portability

Possibilities:

P1 Support for Partial Reconfiguration of Logic

Introduction and Motivation

Long-Term Maintenance, Dependability, Compositionality

Demands:

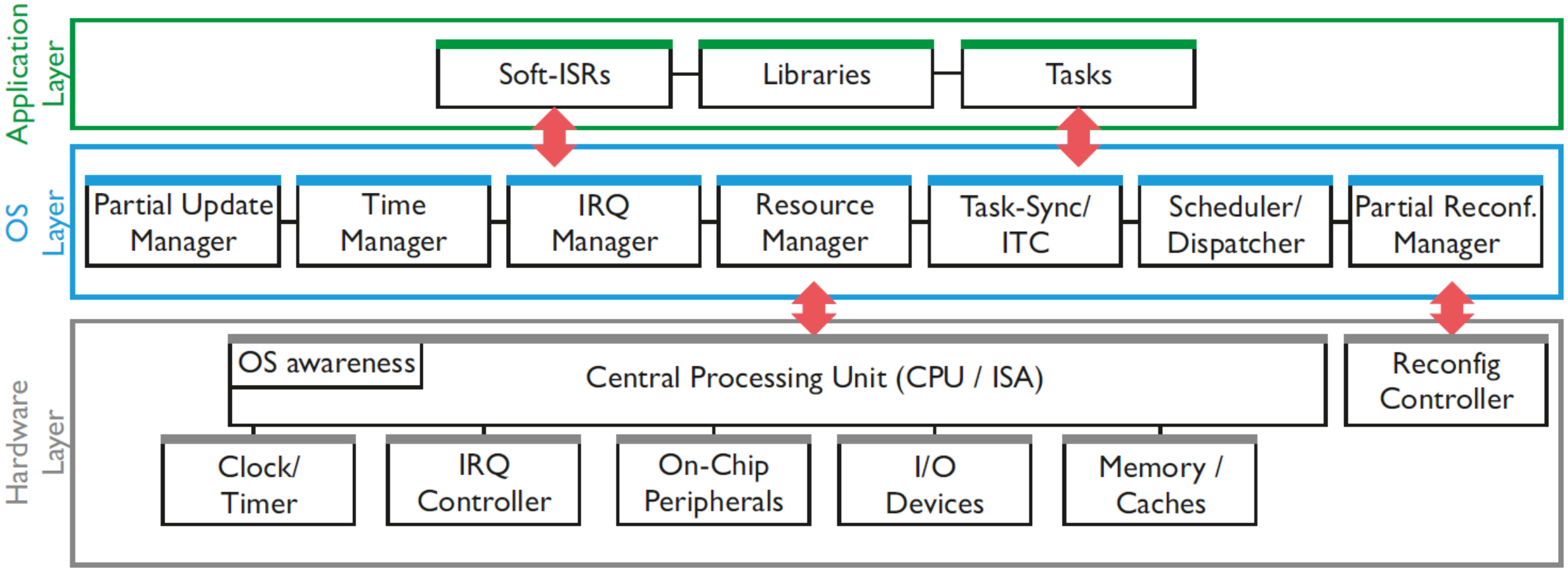
D1 Partial Software Updates
D2 Support for Hardware Modification
D3 Automatic Integration
D4 Hard Correctness Guarantees for Composition
D5 Efficient and Automatic Portability

Possibilities:

P1 Support for Partial Reconfiguration of Logic
P2 Use of Formal Methods for Development and Maintenance

OS Architecture and Basic Concepts

Layering.



OS Architecture and Basic Concepts

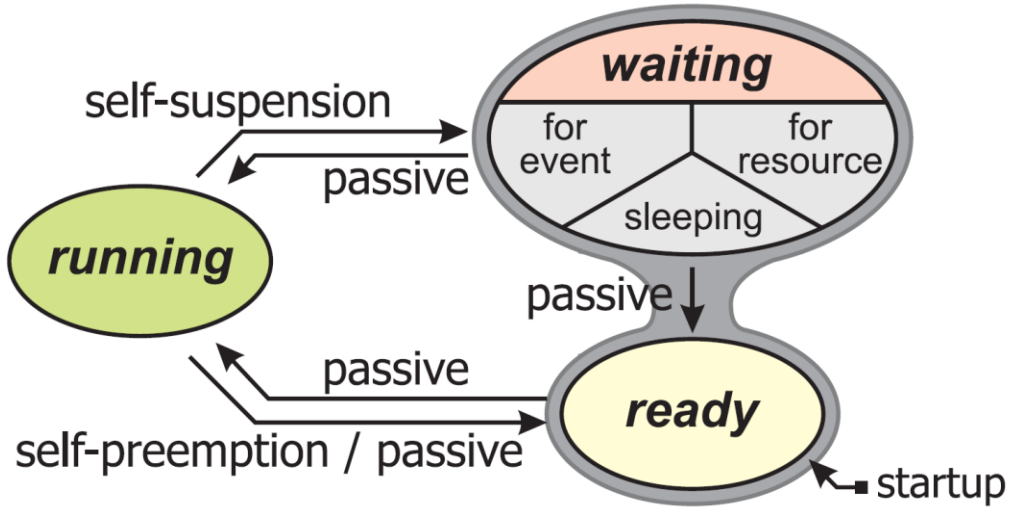
Central Concepts.

- Internal Timeline

OS Architecture and Basic Concepts

Central Concepts.

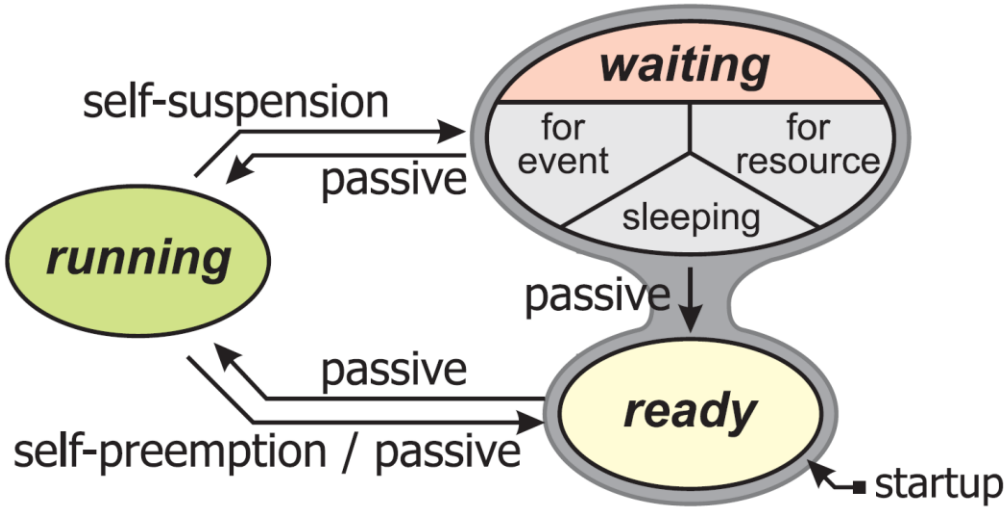
- Internal Timeline
- Tasks



OS Architecture and Basic Concepts

Central Concepts.

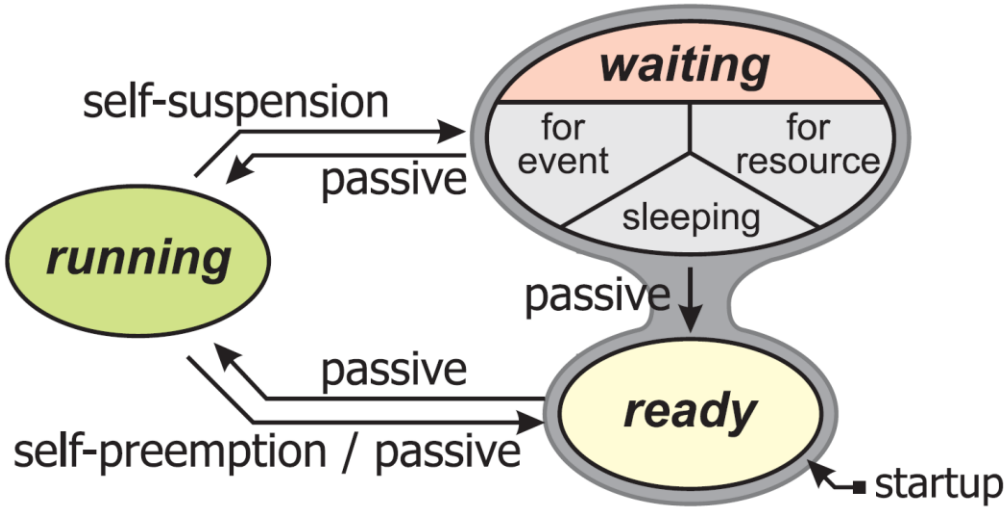
- Internal Timeline
- Tasks
- System Calls



OS Architecture and Basic Concepts

Central Concepts.

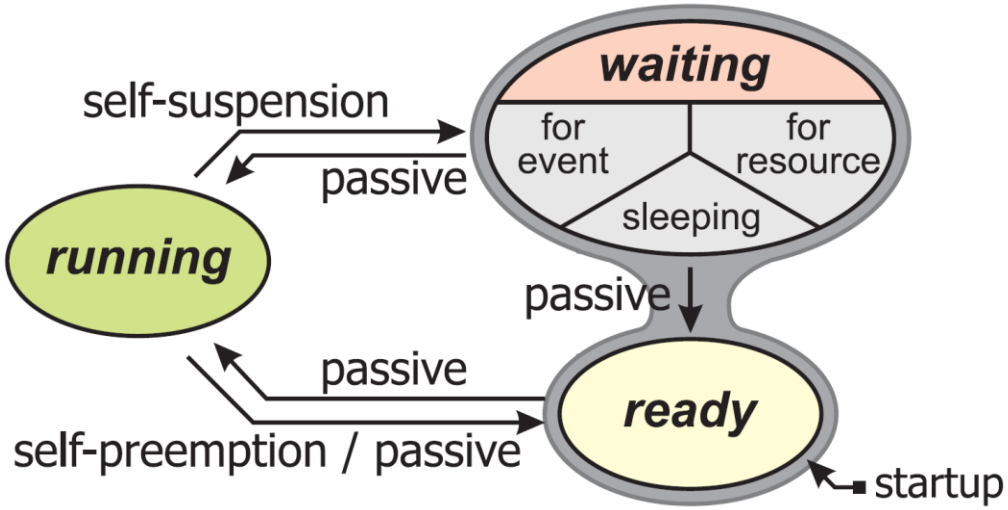
- Internal Timeline
- Tasks
- System Calls
- Events



OS Architecture and Basic Concepts

Central Concepts.

- Internal Timeline
- Tasks
- System Calls
- Events
- Resources

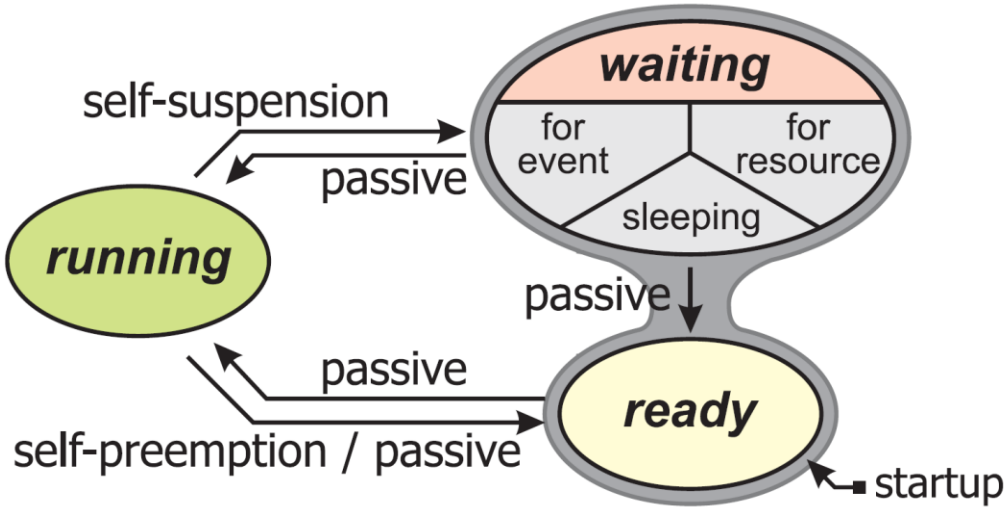



```
1  OS_TASKENTRY (task1) {
2    [...]
3    while (1) {
4      waitEvent (ev1);
5
6      getResource (res1);
7      [...]
8      releaseResource (res1);
9
10     setEvent (ev2);
11   }
12 }
```

OS Architecture and Basic Concepts

Central Concepts.

- Internal Timeline
- Tasks
- System Calls
- Events
- Resources
- Interrupts




```
1  OS_TASKENTRY (task1) {
2    [...]
3    while (1) {
4      waitEvent (ev1);
5
6      getResource (res1);
7      [...]
8      releaseResource (res1);
9
10     setEvent (ev2);
11   }
12 }
```

Extended Concepts and Special Features

An overview.

- Tackling Demands D1-D5

Extended Concepts and Special Features

An overview.

- Tackling Demands D1-D5

- Three overarching Topics:
  - OS-specific Hardware Support and Reconfiguration
  - Compositional Software Design and Partial Updates
  - Formal Methods for Verification and Portability

Extended Concepts and Special Features

An overview.

- Tackling Demands D1-D5

- Three overarching Topics:
  - OS-specific Hardware Support and Reconfiguration
  - Compositional Software Design and Partial Updates
  - Formal Methods for Verification and Portability

- Extended concepts facilitate/improve addressing demands!

Extended Concepts and Special Features

MCU/OS Co-Design.

- **Partial Reconfiguration**
  of the host computing platform at runtime

```
10    [...]
11    addi   t1, zero, 6
12    cinsi  t0, t1,    2    ; unknown
          instr.
13    [...]
```

Extended Concepts and Special Features

MCU/OS Co-Design.

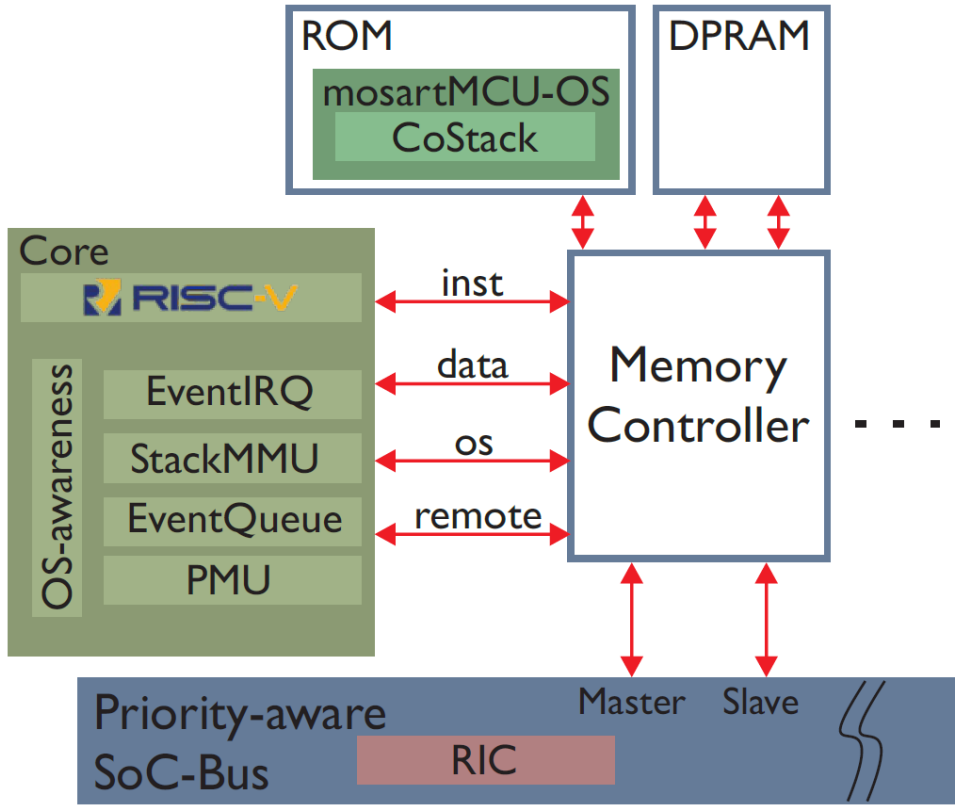
- **Partial Reconfiguration**
  of the host computing platform at runtime
- **Hardware Security Features**
  within the host microcontroller unit

```
10    [...]
11    addi  t1, zero, 6
12    cinsi t0, t1,   2   ; unknown
          instr.
13    [...]
```

Extended Concepts and Special Features

MCU/OS Co-Design.

- **Partial Reconfiguration**
  of the host computing platform at runtime
- **Hardware Security Features**
  within the host microcontroller unit
- **OS-aware Logic**
  of the host processor

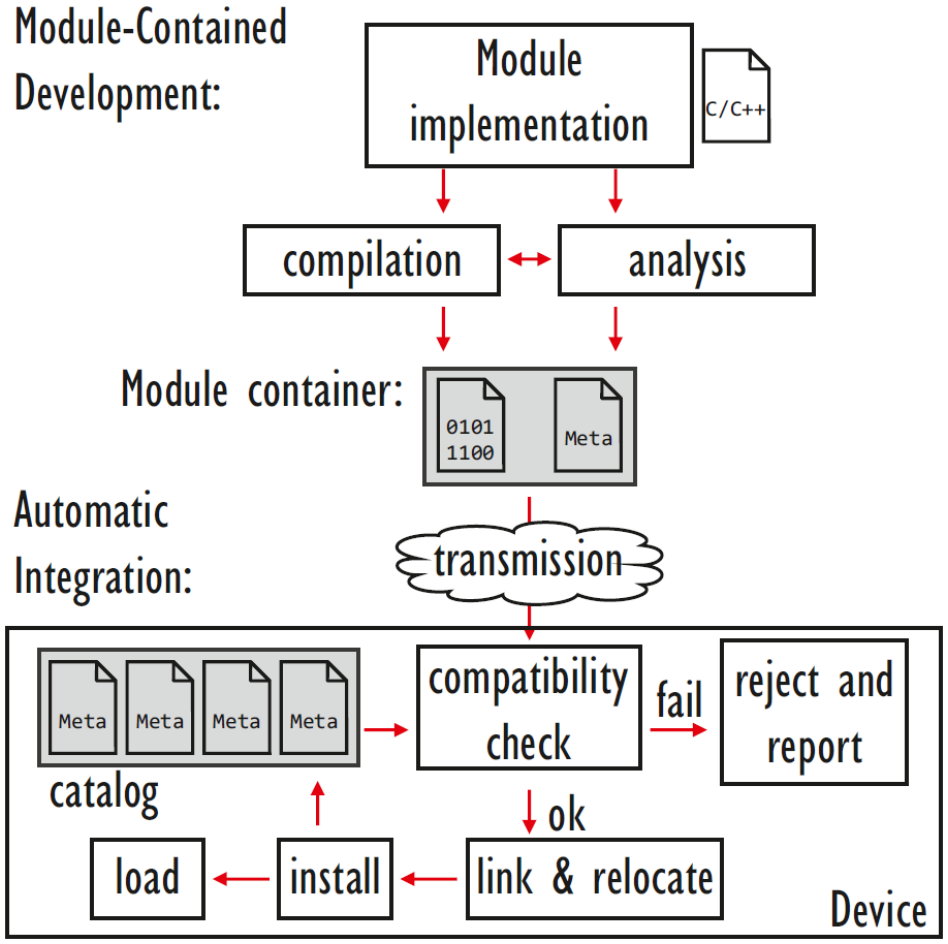



```
10    [...]
11    addi  t1, zero, 6
12    cinsi t0, t1,   2    ; unknown
          instr.
13    [...]
```

Extended Concepts and Special Features

Compositional Software and Automatic Integration.

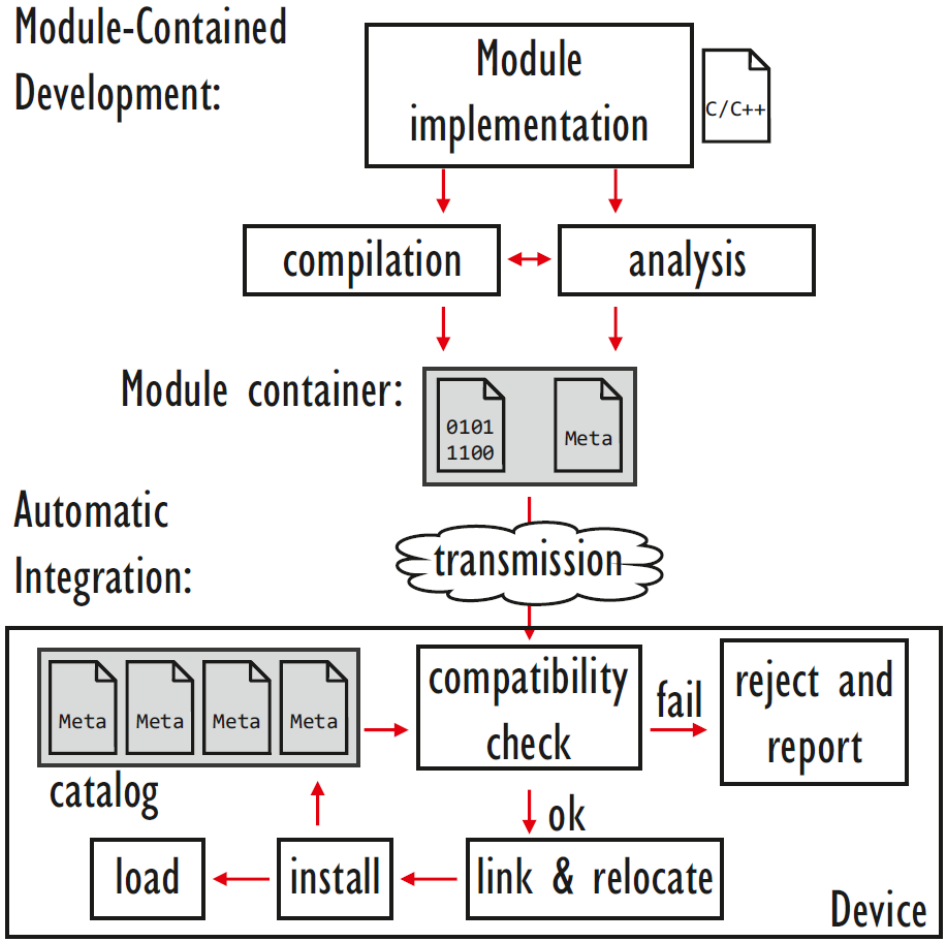
- **Partial Updates**
  of the running software in a modular way
  while preserving code dependencies



Extended Concepts and Special Features

Compositional Software and Automatic Integration.

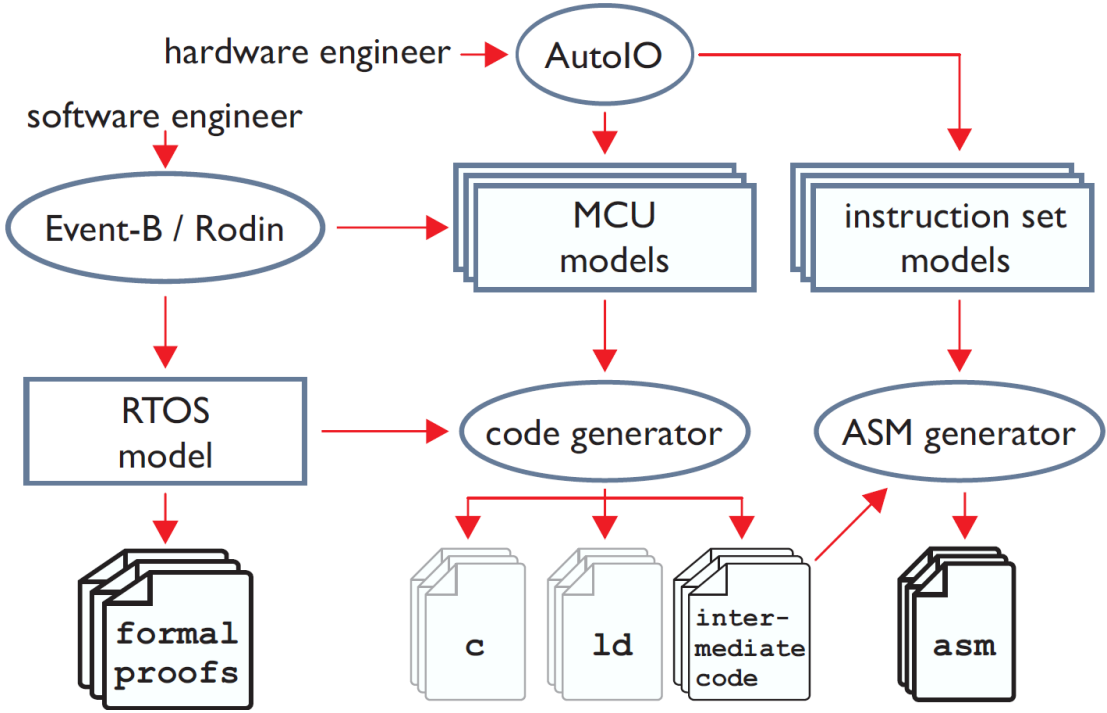
- **Partial Updates**
  of the running software in a modular way
  while preserving code dependencies
- **Compatibility Checks**
  of functional and non-functional requirements
  before an update is applied



Extended Concepts and Special Features

Formal Methods for Verification and Portability.

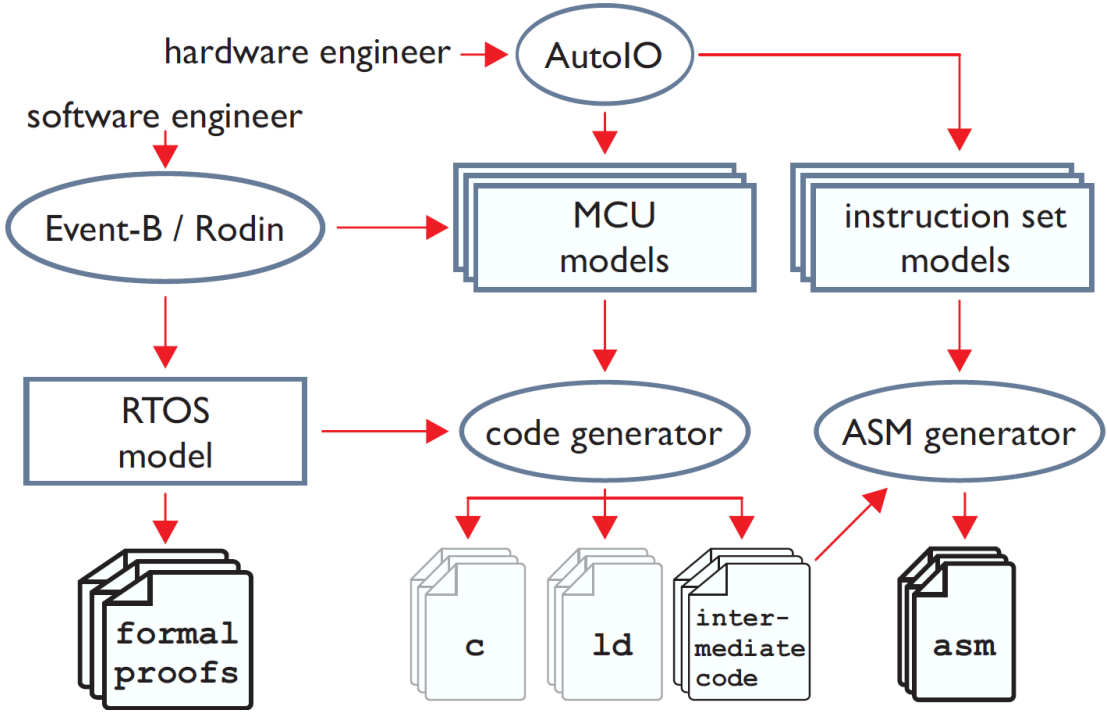
- **Creation of Independent Models**
  of application software, operating system and processor logic



Extended Concepts and Special Features

Formal Methods for Verification and Portability.

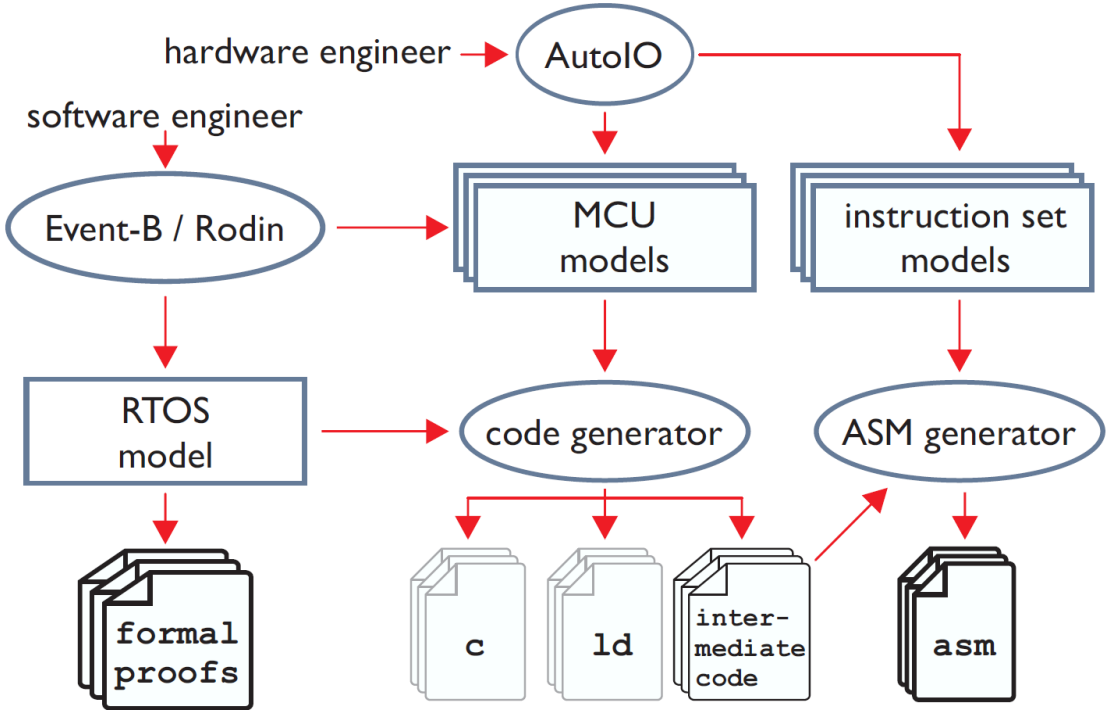
- **Creation of Independent Models**
  of application software, operating system and processor logic
- **Generation**
  of hardware-specific code for different target architectures

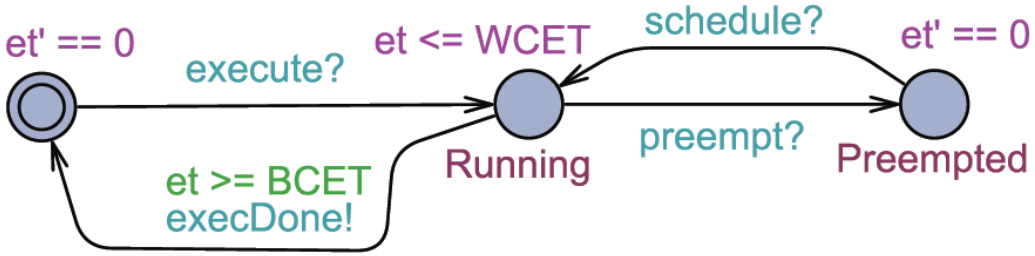


Extended Concepts and Special Features

Formal Methods for Verification and Portability.

- **Creation of Independent Models**
  of application software, operating system and processor logic
- **Generation**
  of hardware-specific code for different target architectures
- **Verification**
  of different aspects of (non-)functional properties





Conclusion

- SmartOS consists of a microkernel with basic concepts

Conclusion

- SmartOS consists of a microkernel with basic concepts
- Extended features include

Conclusion

- SmartOS consists of a microkernel with basic concepts

- Extended features include
  - a tightly coupled design of the OS and its underlying MCU,

Conclusion

- SmartOS consists of a microkernel with basic concepts

- Extended features include
  - a tightly coupled design of the OS and its underlying MCU,
  - support for compositional software, and

Conclusion

- SmartOS consists of a microkernel with basic concepts

- Extended features include
  - a tightly coupled design of the OS and its underlying MCU,
  - support for compositional software, and
  - the use of formal methods to support software development and maintenance.

Conclusion

- SmartOS consists of a microkernel with basic concepts

- Extended features include
  - a tightly coupled design of the OS and its underlying MCU,
  - support for compositional software, and
  - the use of formal methods to support software development and maintenance.

- Raise awareness for the necessity of the shown concepts

Conclusion

- SmartOS consists of a microkernel with basic concepts

- Extended features include
  - a tightly coupled design of the OS and its underlying MCU,
  - support for compositional software, and
  - the use of formal methods to support software development and maintenance.

- Raise awareness for the necessity of the shown concepts
- Providing an OS architecture that inherently supports them by design

Thank you for your attention

https://iti.tugraz.at/research/research-areas/embedded-automotive-systems/