# Reconfigurable Communication Middleware for FlexRay-based Distributed Embedded Systems

Diptesh Majumdar,[1] Licong Zhang,[2] Purandar Bhaduri,[1] Samarjit Chakraborty[2]

[1] *Department of Computer Science and Engineering, IIT Guwahati, India*

[2] *Institute for Real-Time Computer Systems, TU Munich, Germany*

RTCSA 2015

Aug. 20, 2015, Hong Kong

IIT Guwahati

Technische Universität München

# Overview

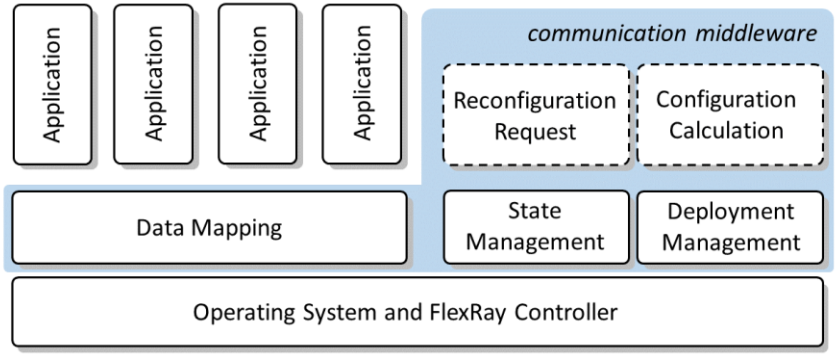
- **Motivation**
  - Automotive E/E architecture increases rapidly in scale and complexity
  - Increasingly more and complex software and data
  - Constrained communication resources
  - Static design and development of safety-critical buses (e.g., FlexRay)
  - Multi-mode, Plug-and-Play applications and software installation/update after sales

# Overview

- **Motivation**
  - Automotive E/E architecture increases rapidly in scale and complexity
  - Increasingly more and complex software and data
  - Constrained communication resources
  - Static design and development of safety-critical buses (e.g., FlexRay)
  - Multi-mode, Plug-and-Play applications and software installation/update after sales
- **Problem**
  - Online communication reconfiguration of FlexRay-based ECU network
  - Communication resource re-allocation for
    - multi-mode applications
    - newly activated applications

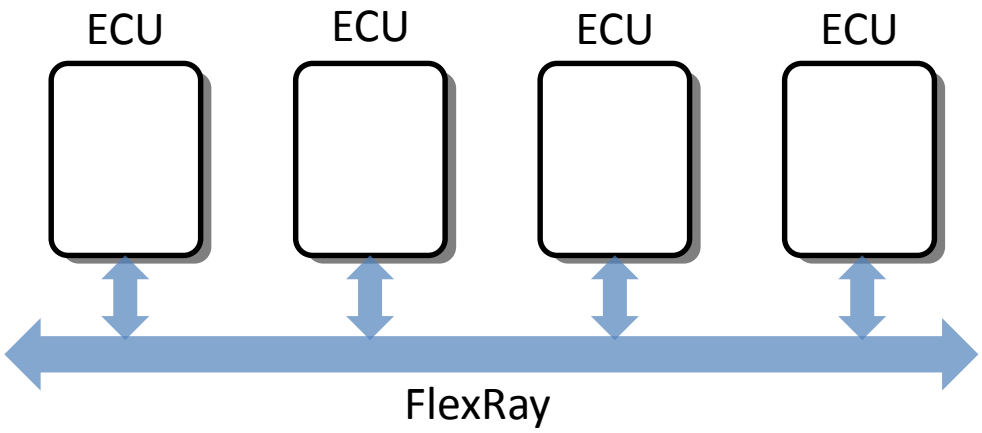
# Overview

- **Motivation**
  - Automotive E/E architecture increases rapidly in scale and complexity
  - Increasingly more and complex software and data
  - Constrained communication resources
  - Static design and development of safety-critical buses (e.g., FlexRay)
  - Multi-mode, Plug-and-Play applications and software installation/update after sales
- **Problem**
  - Online communication reconfiguration of FlexRay-based ECU network
  - Communication resource re-allocation for
    - multi-mode applications
    - newly activated applications
- **Approach**
  - Insertion of a middleware layer
  - Reconfigurable data-to-schedule mapping
  - Online configuration calculation and deployment



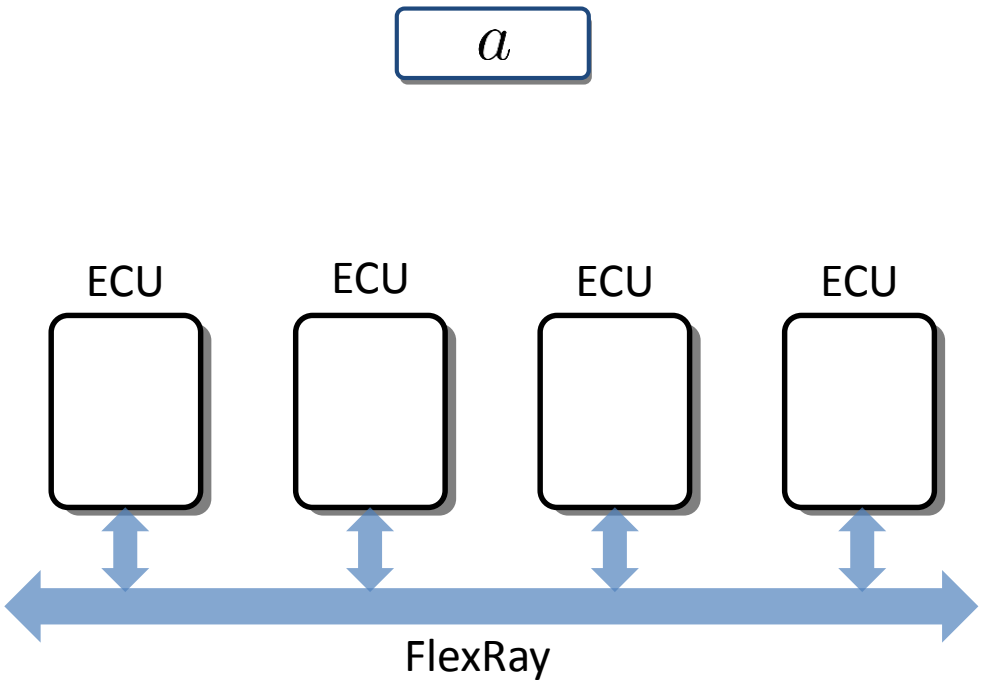
# Background

- **FlexRay-based ECU networks**
  - Hardware architecture



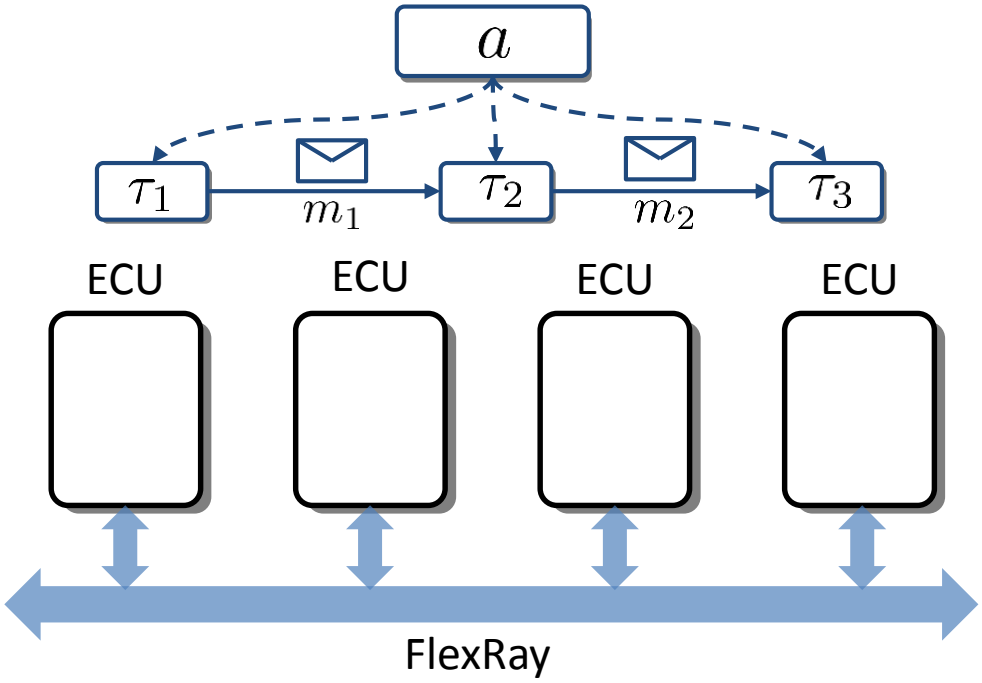
# Background

- **FlexRay-based ECU networks**
  - Hardware architecture
  - Distributed applications



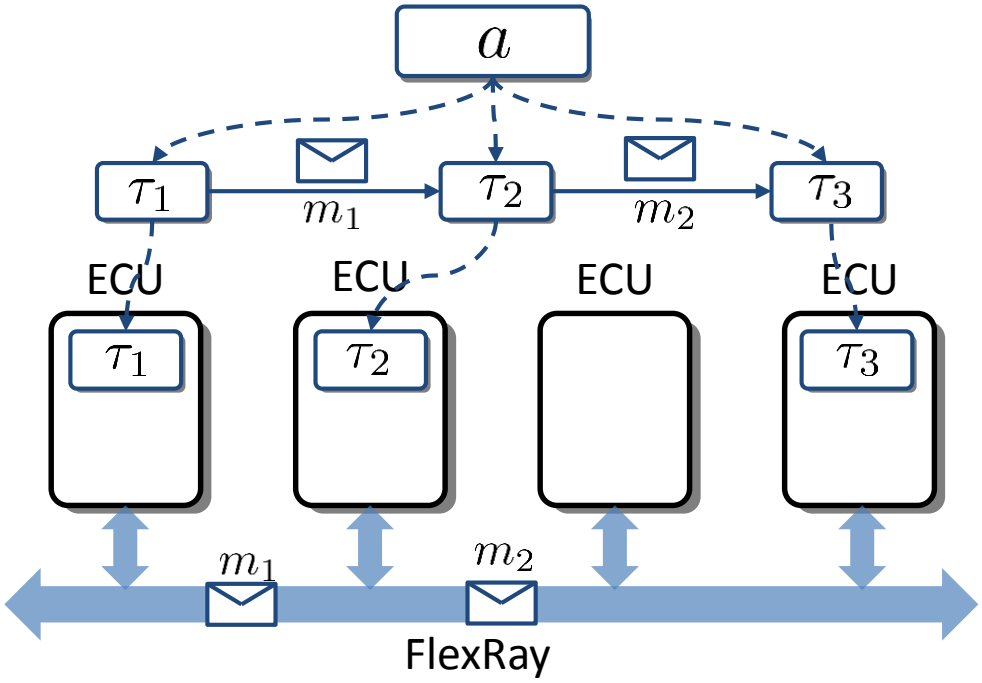
# Background

- **FlexRay-based ECU networks**
  - Hardware architecture
  - Distributed applications



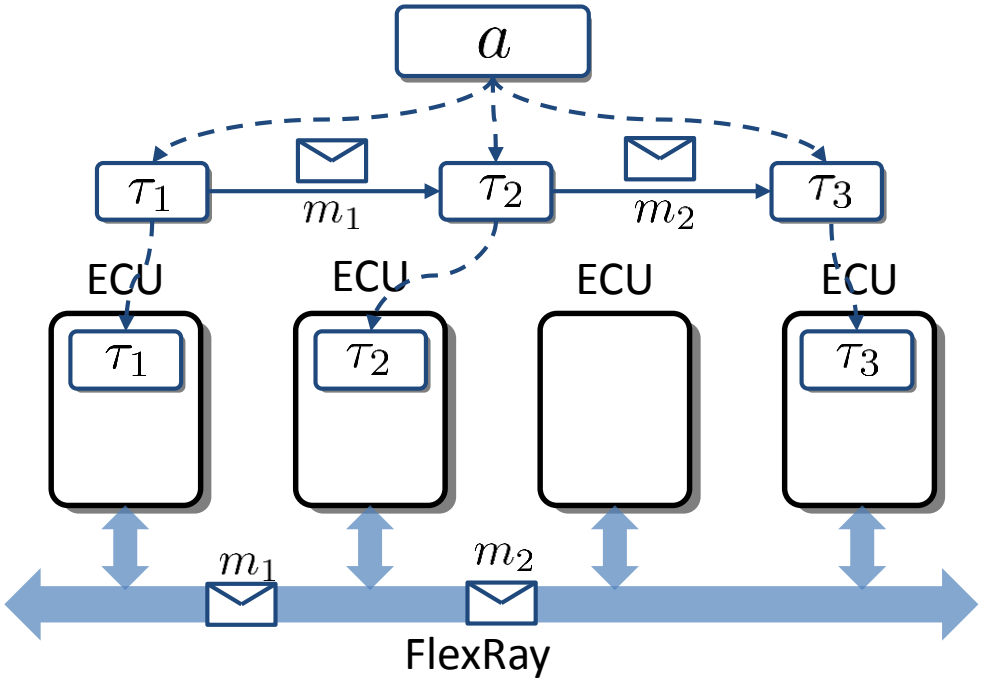
# Background

- **FlexRay-based ECU networks**
  - Hardware architecture
  - Distributed applications
  - Task mapping / bus communication

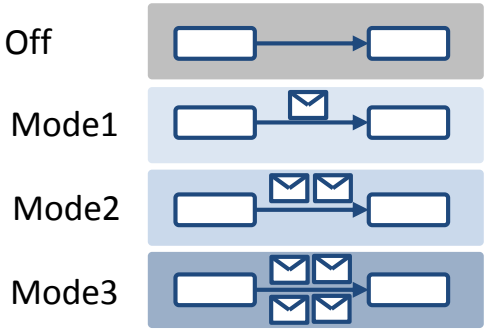


# Background

- **FlexRay-based ECU networks**
  - Hardware architecture
  - Distributed applications
  - Task mapping / bus communication
  - Multi-mode application

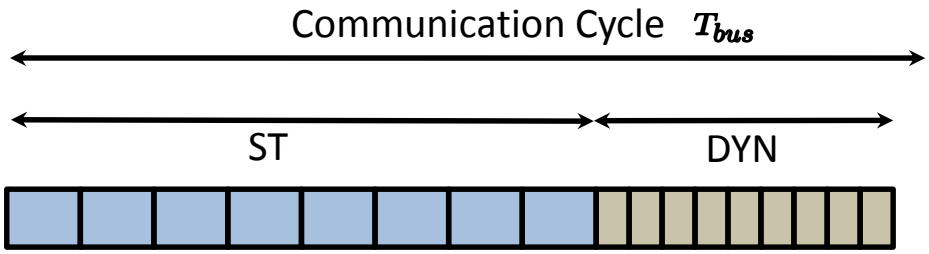



| Mode | Resource | Performance |
|---|---|---|
| 1 | high | high |
| 2 | medium | medium |
| 3 | low | low |
| Off | none | none |



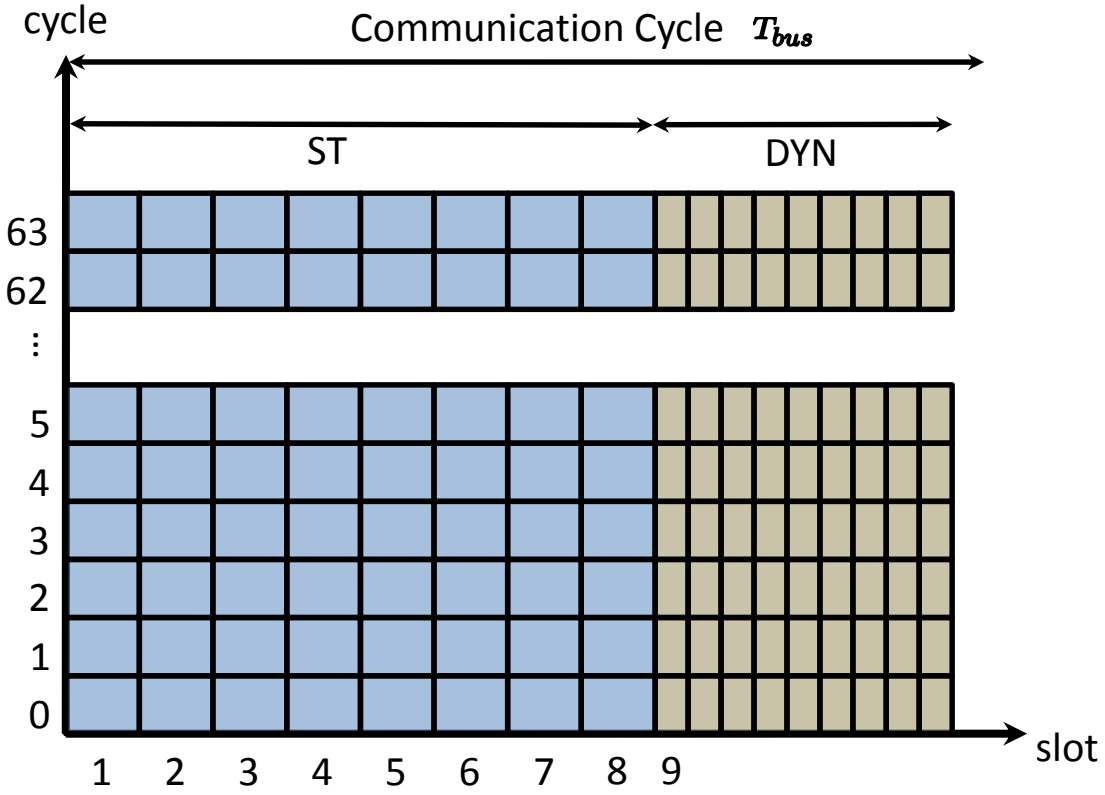
# Background

- **FlexRay communication**
  - Hybrid protocol: mixed time-triggered and event-triggered paradigm
  - Communication cycle, static segment (ST) and dynamic segment (DYN)



# Background

- **FlexRay communication**
  - Hybrid protocol: mixed time-triggered and event-triggered paradigm
  - Communication cycle, static segment (ST) and dynamic segment (DYN)
  - 64-cycle-sequence, FlexRay Schedule $\Theta_i = (S_i, B_i, R_i)$



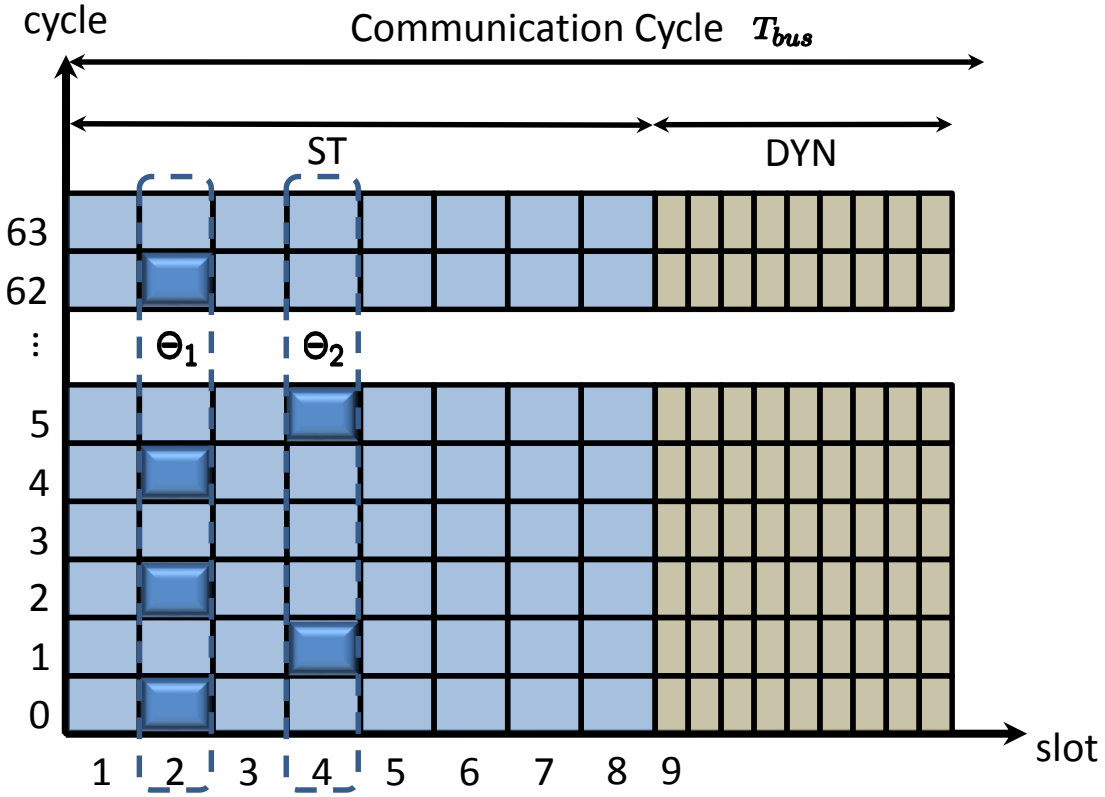

$S_i$ slot number

$B_i$ base cycle

$R_i$ repetition rate

# Background

- **FlexRay communication**
  - Hybrid protocol: mixed time-triggered and event-triggered paradigm
  - Communication cycle, static segment (ST) and dynamic segment (DYN)
  - 64-cycle-sequence, FlexRay Schedule $\Theta_i = (S_i, B_i, R_i)$




$S_i$ slot number

$B_i$ base cycle

$R_i$ repetition rate

$\Theta_1 = (2, 0, 2)$

$\Theta_2 = (4, 1, 4)$

# Motivational Example

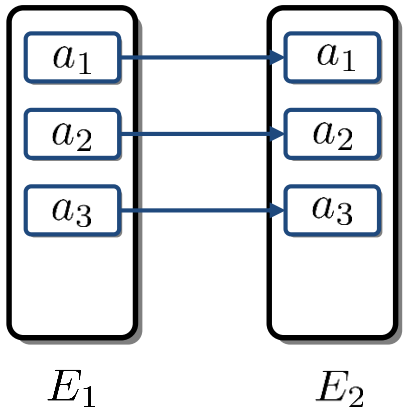
- **Applications**

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2\,T_{bus}$ | 50 |

| $a_2$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 4 | $2\,T_{bus}$ | 100 |
| 2 | 4 | $4\,T_{bus}$ | 80 |
| 3 | 4 | $8\,T_{bus}$ | 50 |

| $a_3$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2T_{bus}$ | 50 |

$\alpha$ mode $w$ data size $p$ period $J$ performance




2 static slots of 8 bytes payload

# Motivational Example

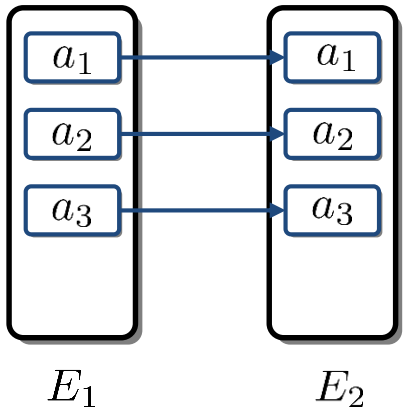
- **Applications**

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2\,T_{bus}$ | 50 |

| $a_2$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 4 | $2\,T_{bus}$ | 100 |
| 2 | 4 | $4\,T_{bus}$ | 80 |
| 3 | 4 | $8\,T_{bus}$ | 50 |

| $a_3$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2T_{bus}$ | 50 |

$\alpha$ mode $w$ data size $p$ period $J$ performance




2 static slots of 8 bytes payload

Possible mode combinations

| | $a_1$ | $a_2$ | $a_3$ | $J$ |
|---|---|---|---|---|
| *1* | 1 | 1 | 2 | 280 |
| *2* | 2 | 1 | 1 | 280 |
| *3* | 1 | 2 | 2 | 260 |
| *4* | 2 | 2 | 1 | 260 |

# Motivational Example

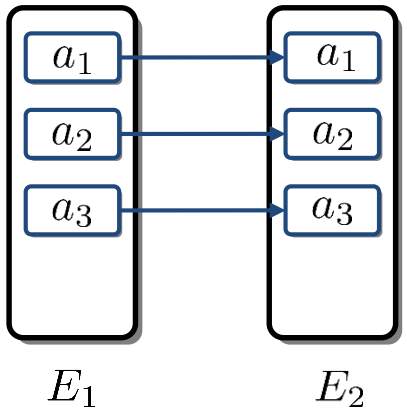
- **Applications**

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2\,T_{bus}$ | 50 |

| $a_2$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 4 | $2\,T_{bus}$ | 100 |
| 2 | 4 | $4\,T_{bus}$ | 80 |
| 3 | 4 | $8\,T_{bus}$ | 50 |

| $a_3$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2T_{bus}$ | 50 |

$\alpha$ mode $w$ data size $p$ period $J$ performance




2 static slots of 8 bytes payload

Possible mode combinations

| | $a_1$ | $a_2$ | $a_3$ | $J$ |
|---|---|---|---|---|
| *1* | 1 | 1 | 2 | 280 |
| *2* | 2 | 1 | 1 | 280 |
| *3* | 1 | 2 | 2 | 260 |
| *4* | 2 | 2 | 1 | 260 |

**online switch not possible**

# Motivational Example

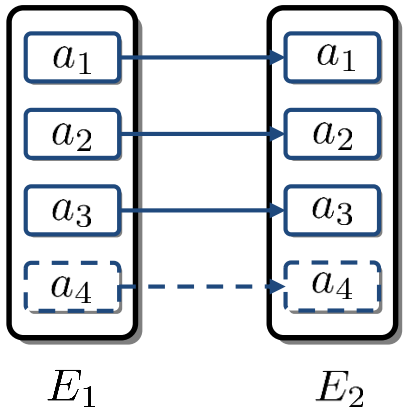
- **Applications**

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2\,T_{bus}$ | 50 |

| $a_2$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 4 | $2\,T_{bus}$ | 100 |
| 2 | 4 | $4\,T_{bus}$ | 80 |
| 3 | 4 | $8\,T_{bus}$ | 50 |

| $a_3$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2T_{bus}$ | 50 |

$\alpha$ mode $w$ data size $p$ period $J$ performance




2 static slots of 8 bytes payload

Possible mode combinations

| | $a_1$ | $a_2$ | $a_3$ | $J$ |
|---|---|---|---|---|
| *1* | 1 | 1 | 2 | 280 |
| *2* | 2 | 1 | 1 | 280 |
| *3* | 1 | 2 | 2 | 260 |
| *4* | 2 | 2 | 1 | 260 |

**online switch not possible**

Newly installed application

| $a_4$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 4 | $2\,T_{bus}$ | 100 |

# Motivational Example

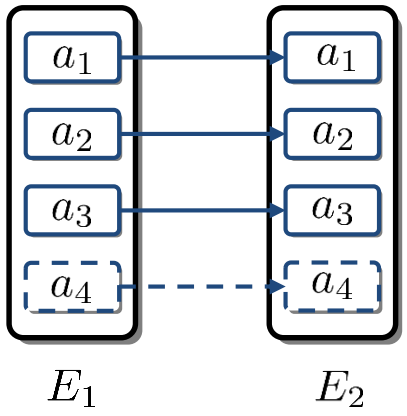
- **Applications**

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2\,T_{bus}$ | 50 |

| $a_2$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 4 | $2\,T_{bus}$ | 100 |
| 2 | 4 | $4\,T_{bus}$ | 80 |
| 3 | 4 | $8\,T_{bus}$ | 50 |

| $a_3$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2T_{bus}$ | 50 |

$\alpha$ mode $w$ data size $p$ period $J$ performance




2 static slots of 8 bytes payload

Possible mode combinations

| | $a_1$ | $a_2$ | $a_3$ | $J$ |
|---|---|---|---|---|
| *1* | 1 | 1 | 2 | 280 |
| *2* | 2 | 1 | 1 | 280 |
| *3* | 1 | 2 | 2 | 260 |
| *4* | 2 | 2 | 1 | 260 |

**online switch not possible**

Newly installed application

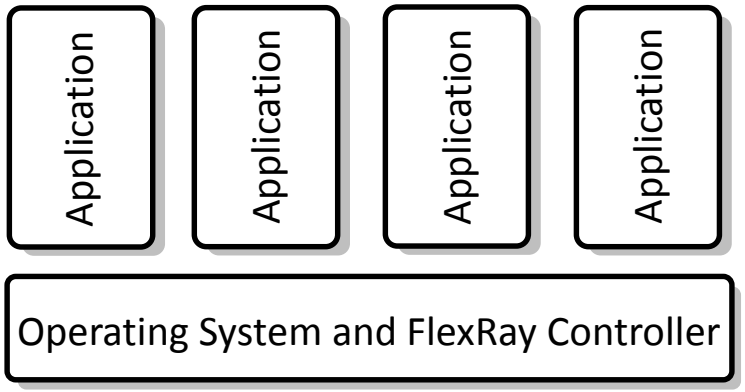
| $a_4$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 4 | $2\,T_{bus}$ | 100 |

**online activation not possible**

although enough resource to map it on combination *1*

# Middleware

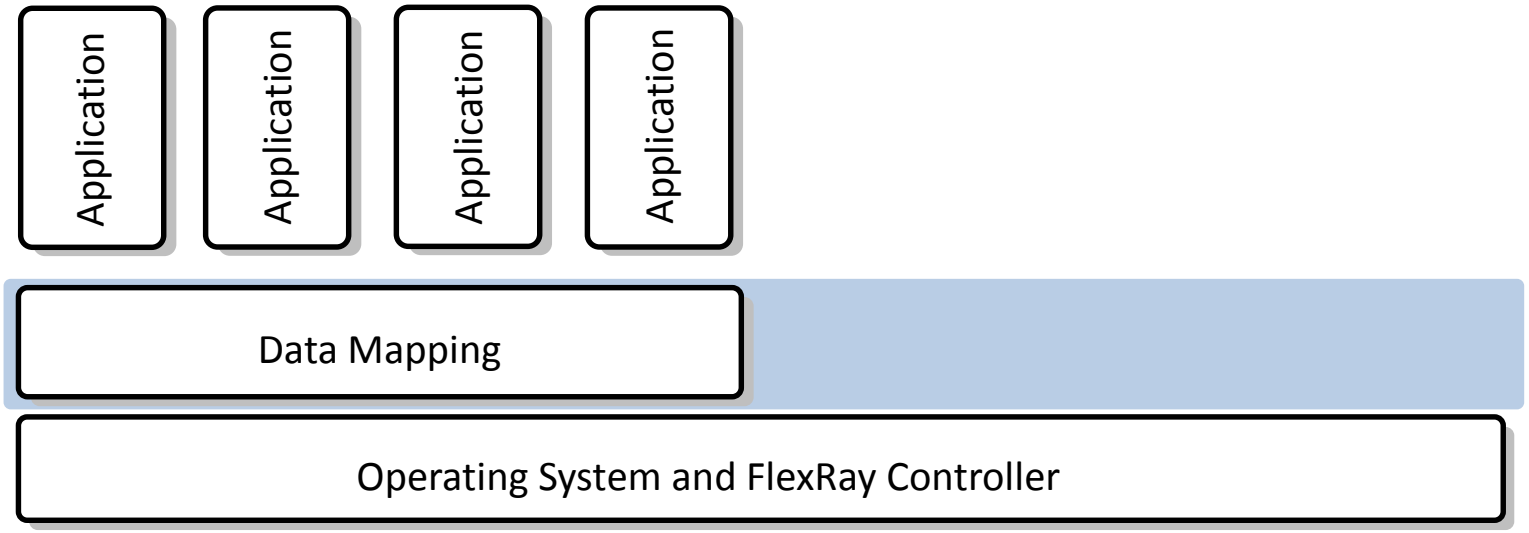
- **Software architecture**
  - Simple software architecture




  - FlexRay parameters, FlexRay schedules and data-to-frame mapping statically configured offline

# Middleware

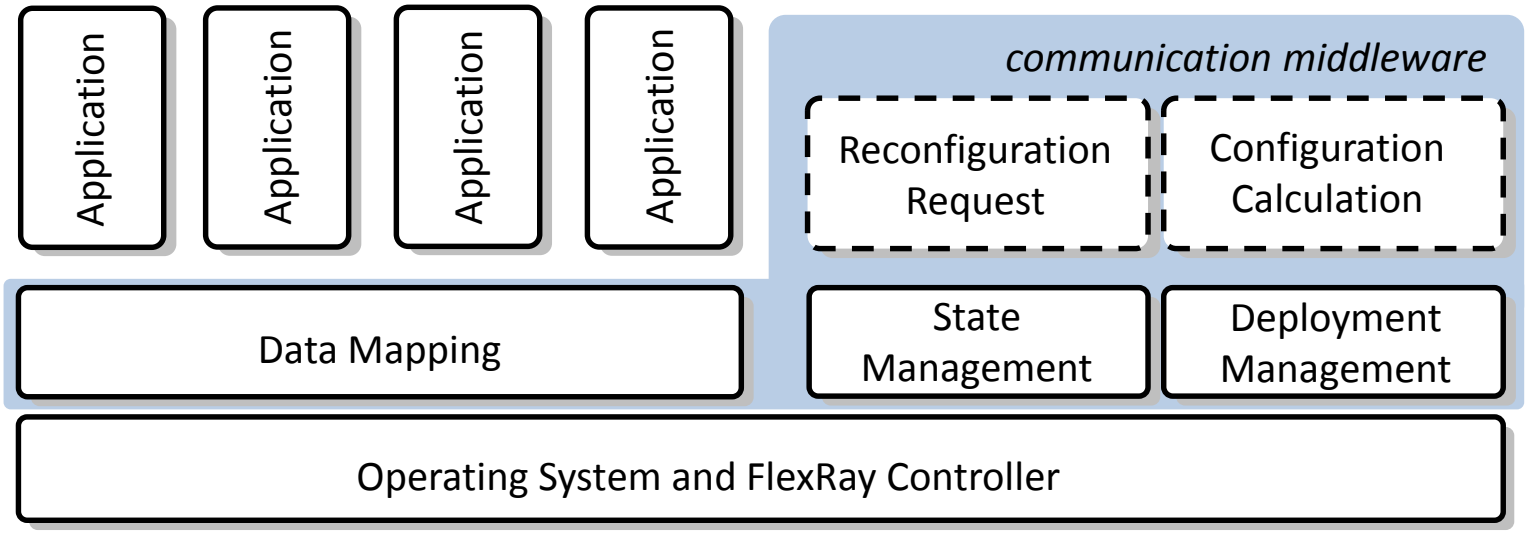
- **Software architecture**
  - Proposed software architecture




  - Middleware between application and OS layers
    - Data mapping
      - Data-to-frame mapping based on a configuration

# Middleware

- **Software architecture**
  - Proposed software architecture




  - Middleware between application and OS layers
    - Data mapping
    - Reconfiguration request
    - Configuration calculator
    - State management
    - Deployment management

# Middleware

- **Application, task, message, mode and manifest**
  - Application $a_i(\alpha_i) = (\mathcal{T}_i(\alpha_i), \mathcal{M}_i(\alpha_i), \mathcal{J}_i(\alpha_i))$
  - Task $\tau_j \in \mathcal{T}_i$ , message $m_j \in \mathcal{M}_i$ , performance $J_i \in \mathcal{J}_i$ , mode $\alpha_i$
  - Characteristics of $\mathcal{T}_i$ , $\mathcal{M}_i$ and $\mathcal{J}_i$ are known and provided by the application developer as application manifest

# Middleware

- **Application, task, message, mode and manifest**
  - Application $a_i(\alpha_i) = (\mathcal{T}_i(\alpha_i), \mathcal{M}_i(\alpha_i), \mathcal{J}_i(\alpha_i))$
  - Task $\tau_j \in \mathcal{T}_i$ , message $m_j \in \mathcal{M}_i$ , performance $J_i \in \mathcal{J}_i$ , mode $\alpha_i$
  - Characteristics of $\mathcal{T}_i$ , $\mathcal{M}_i$ and $\mathcal{J}_i$ are known and provided by the application developer as application manifest
  - An example in terms of communication

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2\,T_{bus}$ | 50 |

# Middleware

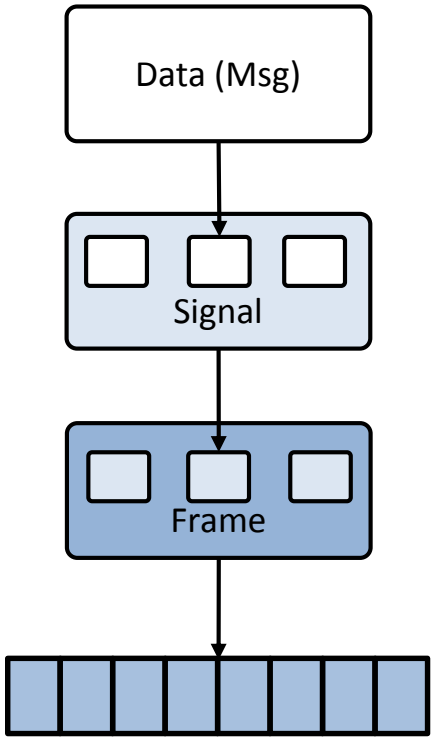
- **Application, task, message, mode and manifest**
  - Application $a_i(\alpha_i) = (\mathcal{T}_i(\alpha_i), \mathcal{M}_i(\alpha_i), \mathcal{J}_i(\alpha_i))$
  - Task $\tau_j \in \mathcal{T}_i$ , message $m_j \in \mathcal{M}_i$ , performance $J_i \in \mathcal{J}_i$ , mode $\alpha_i$
  - Characteristics of $\mathcal{T}_i$ , $\mathcal{M}_i$ and $\mathcal{J}_i$ are known and provided by the application developer as application manifest
  - An example in terms of communication

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $T_{bus}$ | 80 |
| 3 | 4 | $2\,T_{bus}$ | 50 |

- **Configuration**
  - Application configuration $C_a = \{\alpha_i | a_i \in \mathcal{A}\}$
  - Communication configuration $C_c = \{\mathcal{M}_i | a_i \in \mathcal{A}\}$
  - $C_c$ contains the mapping of messages to FlexRay schedule $m_j \rightarrow \Theta_j$

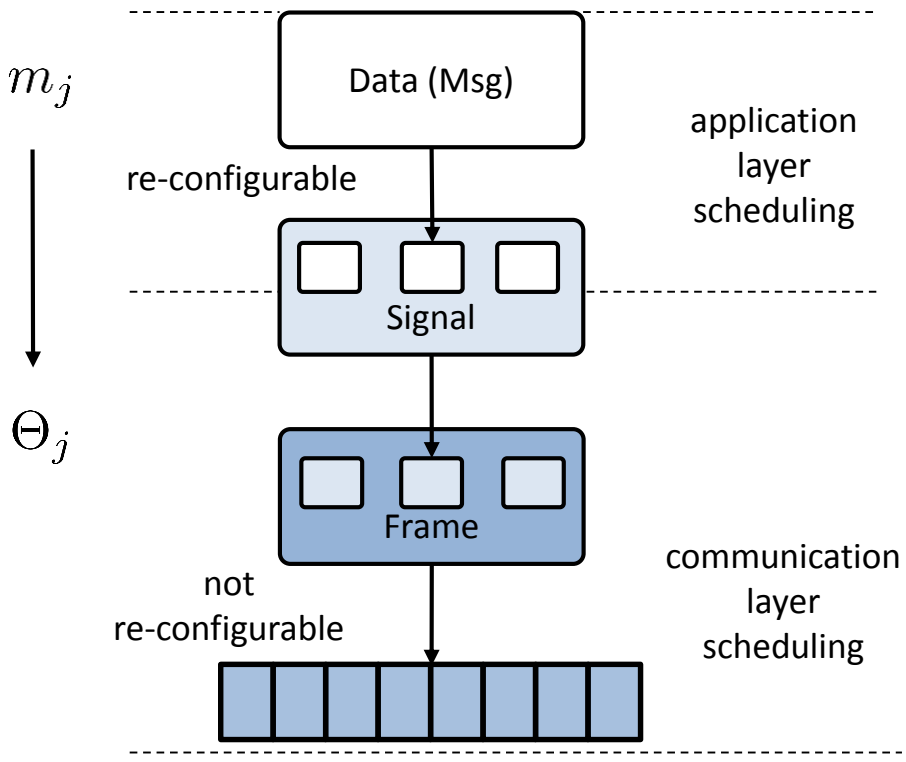
# Middleware

- **Data mapping**



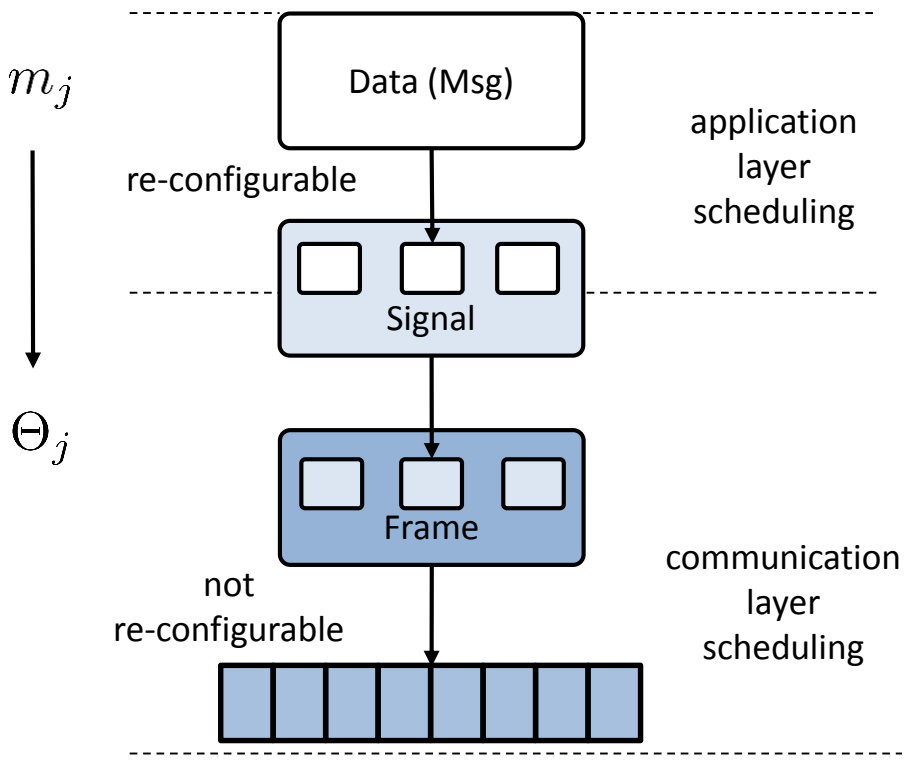
# Middleware

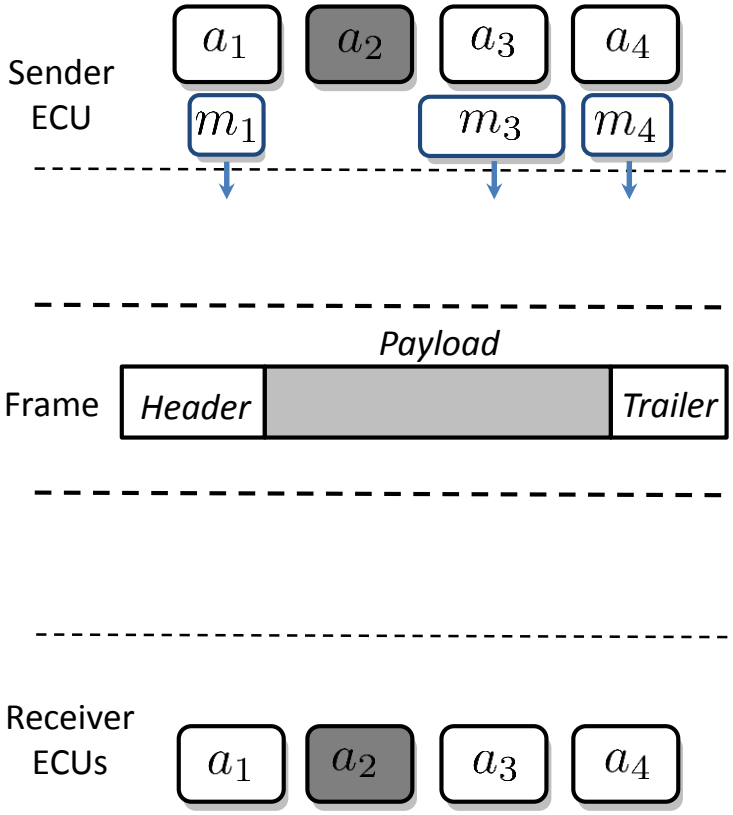
- **Data mapping**
  - For each static slot $S_j$, a FlexRay frame with $\Theta_{j,base} = \{S_j, 0, 1\}$ is assigned with maximal payload possible



# Middleware

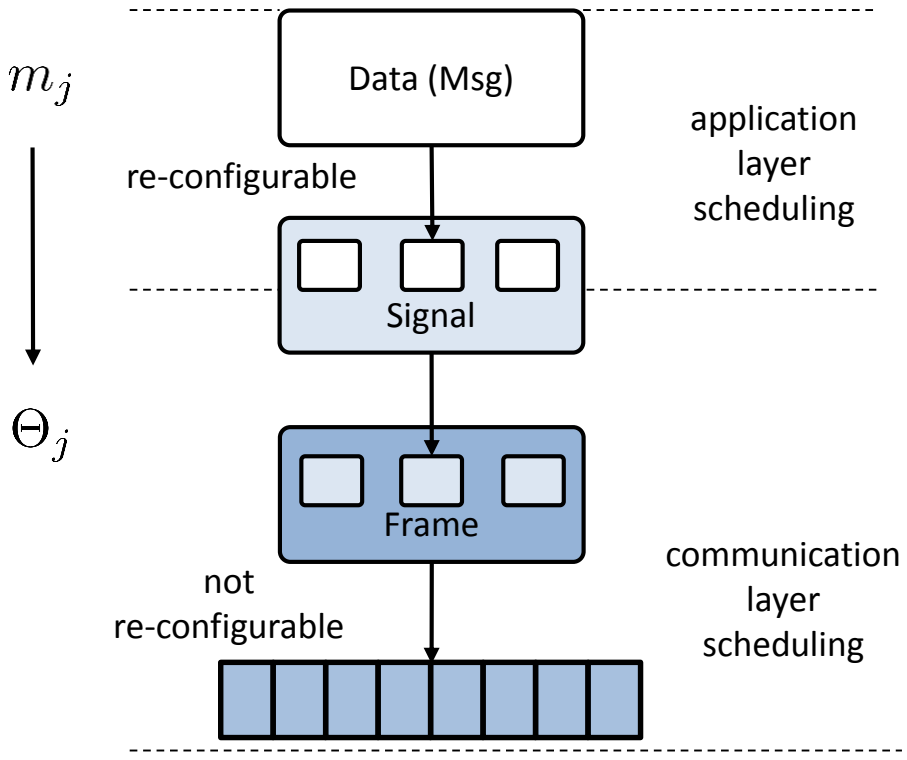
- **Data mapping**
  - For each static slot $S_j$, a FlexRay frame with $\Theta_{j,base} = \{S_j, 0, 1\}$ is assigned with maximal payload possible
  - Data are mapped into $\Theta_{j,base}$ by payload and slot multiplexing within an ECU

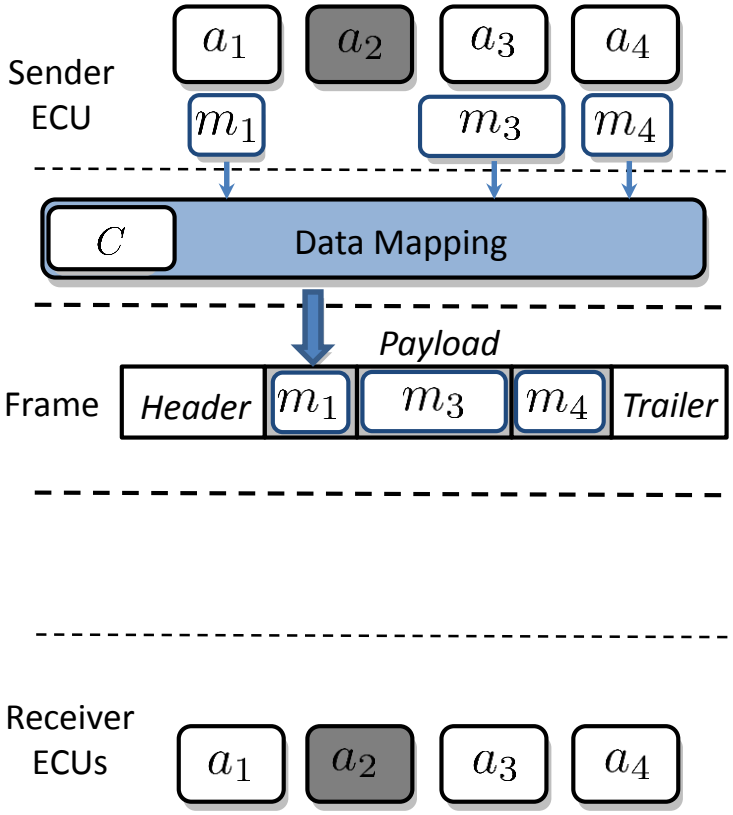


# Middleware

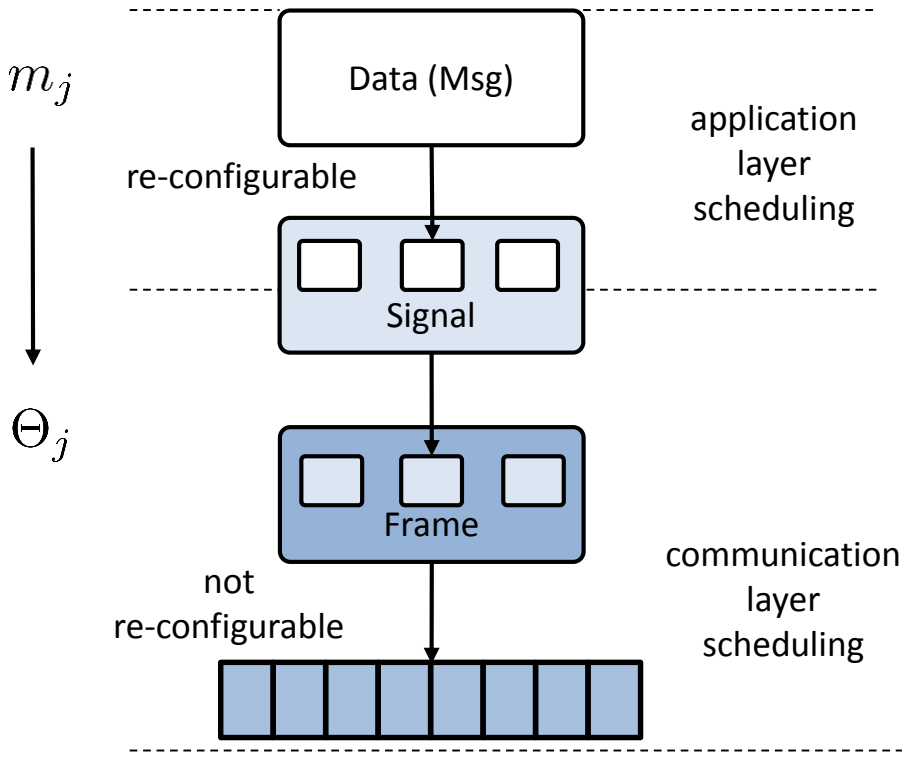
- **Data mapping**
  - For each static slot $S_j$, a FlexRay frame with $\Theta_{j,base} = \{S_j, 0, 1\}$ is assigned with maximal payload possible
  - Data are mapped into $\Theta_{j,base}$ by payload and slot multiplexing within an ECU

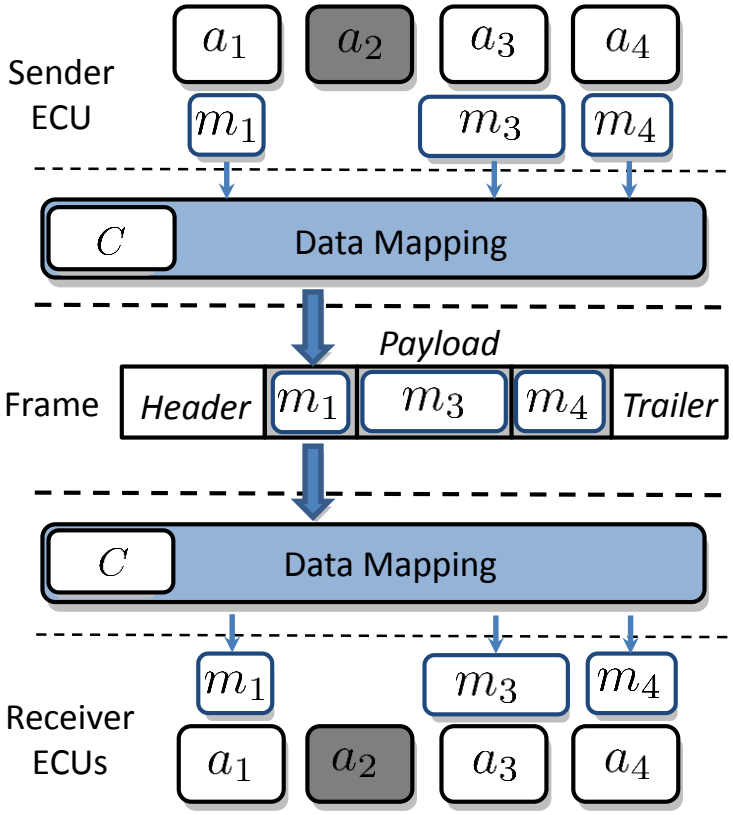


# Middleware

- **Data mapping**
  - For each static slot $S_j$, a FlexRay frame with $\Theta_{j,base} = \{S_j, 0, 1\}$ is assigned with maximal payload possible
  - Data are mapped into $\Theta_{j,base}$ by payload and slot multiplexing within an ECU



# Middleware

- **Re-configuration request**
  - Generates a request for reconfiguration
  - Can be triggered by pre-programmed sequence, application requiring mode switch or activation of newly installed application

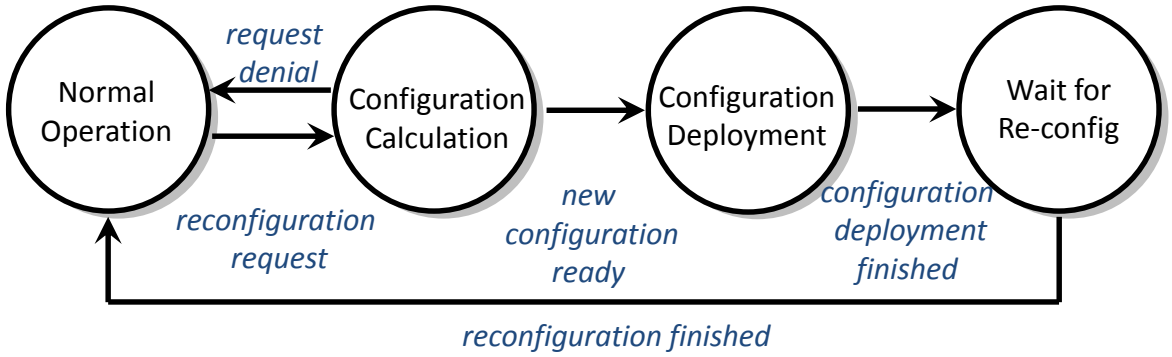
# Middleware

- **Reconfiguration request**
  - Generates a request for reconfiguration
  - Can be triggered by pre-programmed sequence, application requiring mode switch or activation of newly installed application

- **Configuration calculation**
  - To synthesize configuration $C_a = \{\alpha_i | a_i \in \mathcal{A}\}$ and $C_c = \{\mathcal{M}_i | a_i \in \mathcal{A}\}$, where the request and constraints are satisfied and overall performance maximized
  - Can be divided into a two layer problem
    - Layer one: all possible combinations of application modes are traversed
    - Layer two: to determine a feasible $C_c$
      - An exhaustive search
      - A classical bin packing ILP formulation for FlexRay slot packing [8]

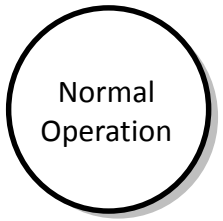
# Middleware

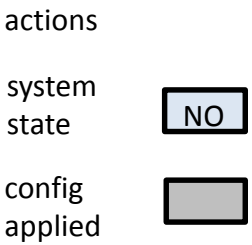
- **Configuration deployment and state management**



# Middleware

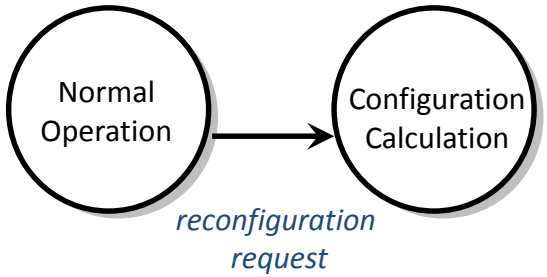
- **Configuration deployment and state management**

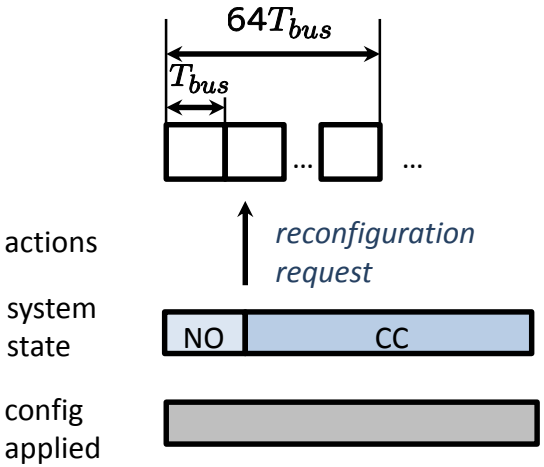


# Middleware

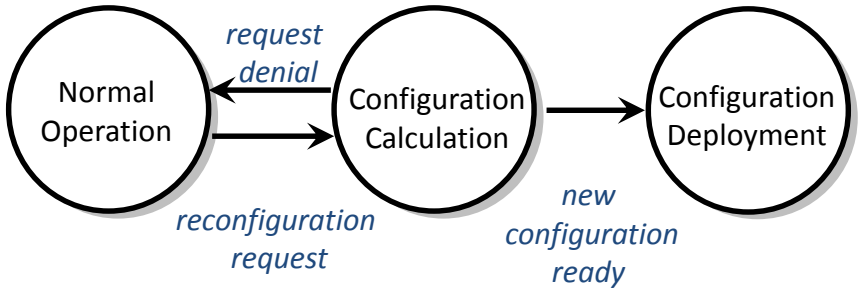
- **Configuration deployment and state management**

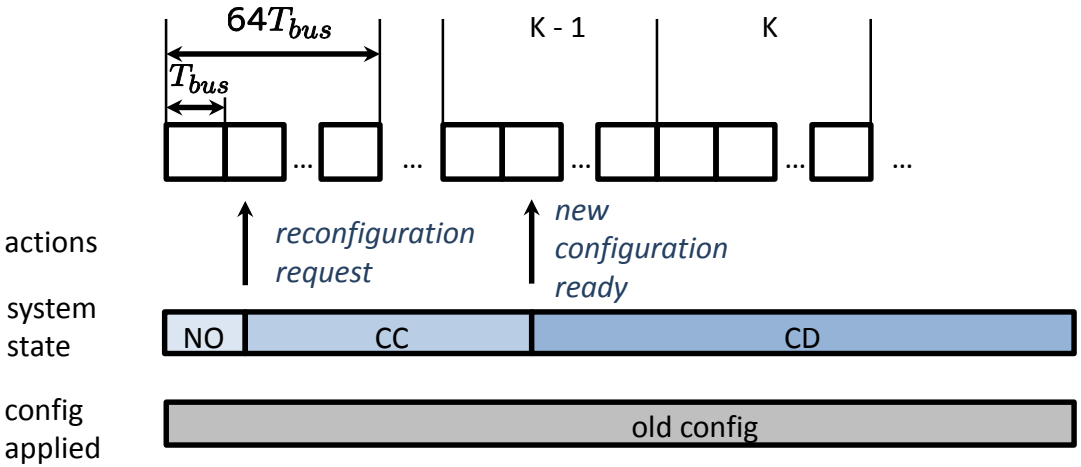


# Middleware

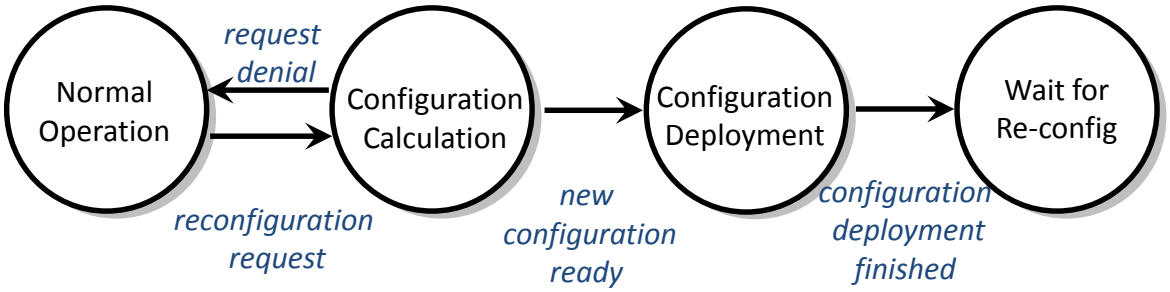
- **Configuration deployment and state management**

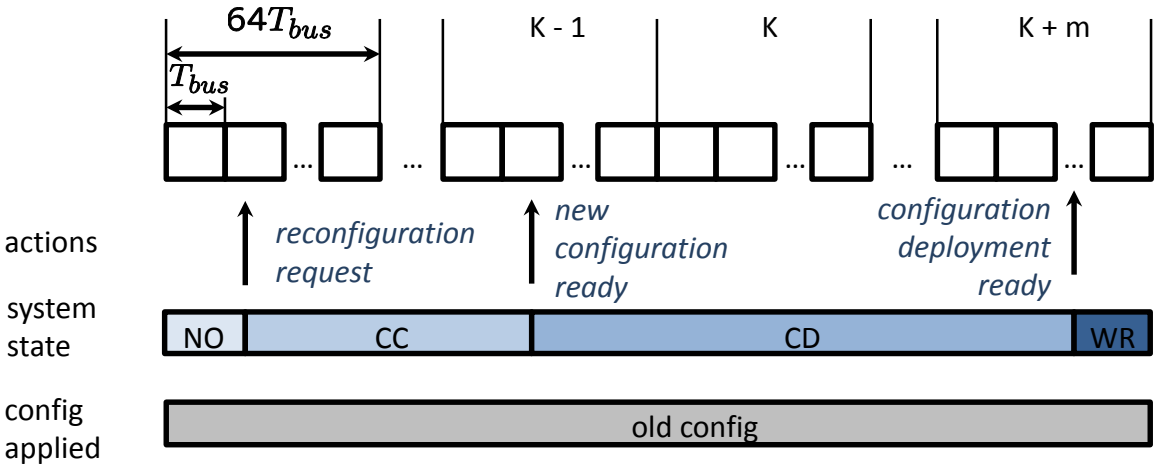


# Middleware

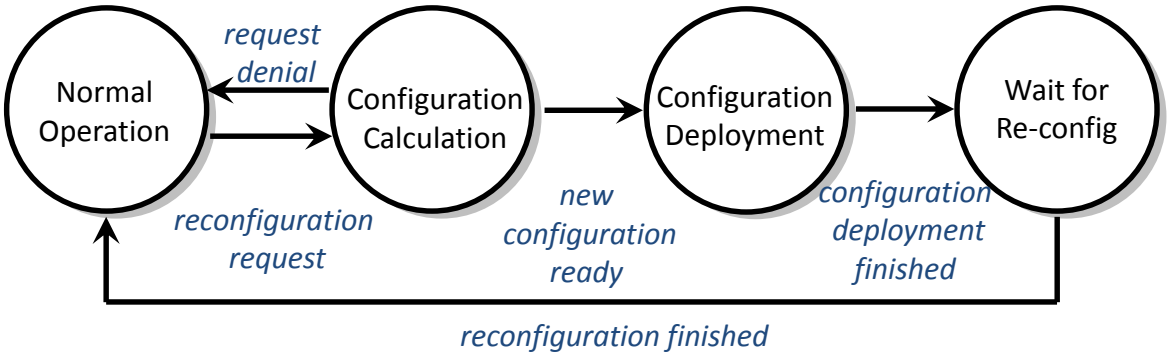
- **Configuration deployment and state management**

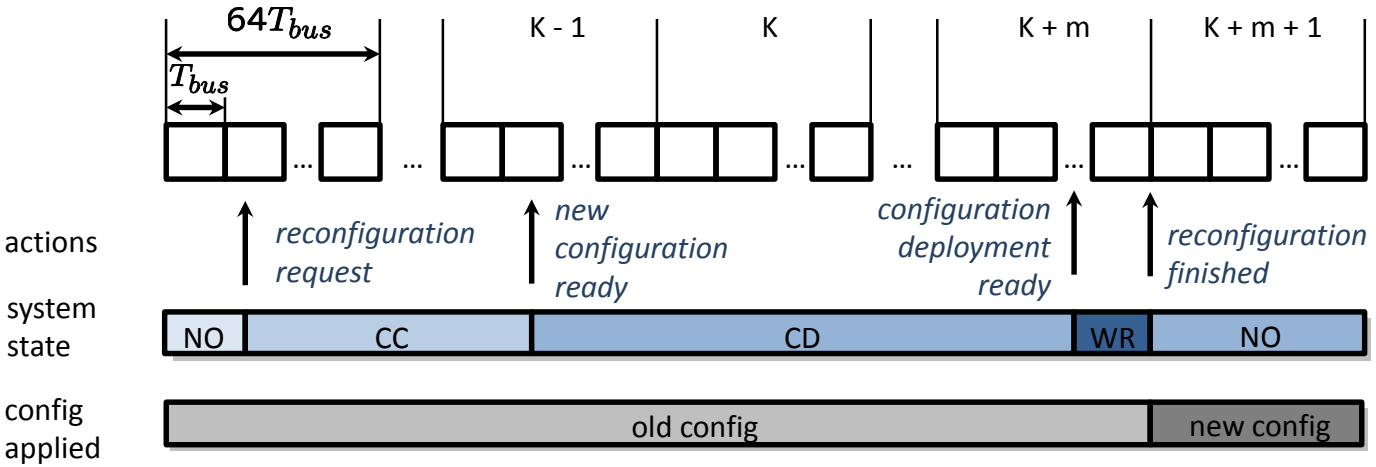


# Middleware

- **Configuration deployment and state management**



# Case Study

- **System Description**
  - 2 ECUs, 4 applications

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 16 | $T_{bus}$ | 100 |
| 2 | 16 | $2\,T_{bus}$ | 50 |
| 3 | 8 | $2\,T_{bus}$ | 25 |

| $a_2$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 8 | $2\,T_{bus}$ | 56 |
| 3 | 8 | $4\,T_{bus}$ | 34 |

| $a_3$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 16 | $T_{bus}$ | 100 |
| 2 | 8 | $T_{bus}$ | 50 |
| 3 | 4 | $T_{bus}$ | 25 |

| $a_4$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $2\,T_{bus}$ | 56 |
| 3 | 8 | $4\,T_{bus}$ | 34 |

# Case Study

- **System Description**
  - 2 ECUs, 4 applications

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 16 | $T_{bus}$ | 100 |
| 2 | 16 | $2\,T_{bus}$ | 50 |
| 3 | 8 | $2\,T_{bus}$ | 25 |

| $a_2$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 8 | $2\,T_{bus}$ | 56 |
| 3 | 8 | $4\,T_{bus}$ | 34 |

| $a_3$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 16 | $T_{bus}$ | 100 |
| 2 | 8 | $T_{bus}$ | 50 |
| 3 | 4 | $T_{bus}$ | 25 |

| $a_4$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $2\,T_{bus}$ | 56 |
| 3 | 8 | $4\,T_{bus}$ | 34 |

- **Reconfiguration requests**

| **step** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **request** | $a_1 \rightarrow \text{on}$ | $a_2 \rightarrow \text{on}$ | $a_3 \rightarrow \text{on}$ | $\alpha_3 \rightarrow 1$ | $a_4 \rightarrow \text{on}$ | $\alpha_4 \rightarrow 1$ | $a_3 \rightarrow \text{off}$ | $a_3 \rightarrow \text{on}$ |

# Case Study

- **System Description**
  - 2 ECUs, 4 applications

| $a_1$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 16 | $T_{bus}$ | 100 |
| 2 | 16 | $2\,T_{bus}$ | 50 |
| 3 | 8 | $2\,T_{bus}$ | 25 |

| $a_2$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 8 | $2\,T_{bus}$ | 56 |
| 3 | 8 | $4\,T_{bus}$ | 34 |

| $a_3$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 16 | $T_{bus}$ | 100 |
| 2 | 8 | $T_{bus}$ | 50 |
| 3 | 4 | $T_{bus}$ | 25 |

| $a_4$ | | | |
|---|---|---|---|
| $\alpha$ | $w$ | $p$ | $J$ |
| 1 | 8 | $T_{bus}$ | 100 |
| 2 | 4 | $2\,T_{bus}$ | 56 |
| 3 | 8 | $4\,T_{bus}$ | 34 |

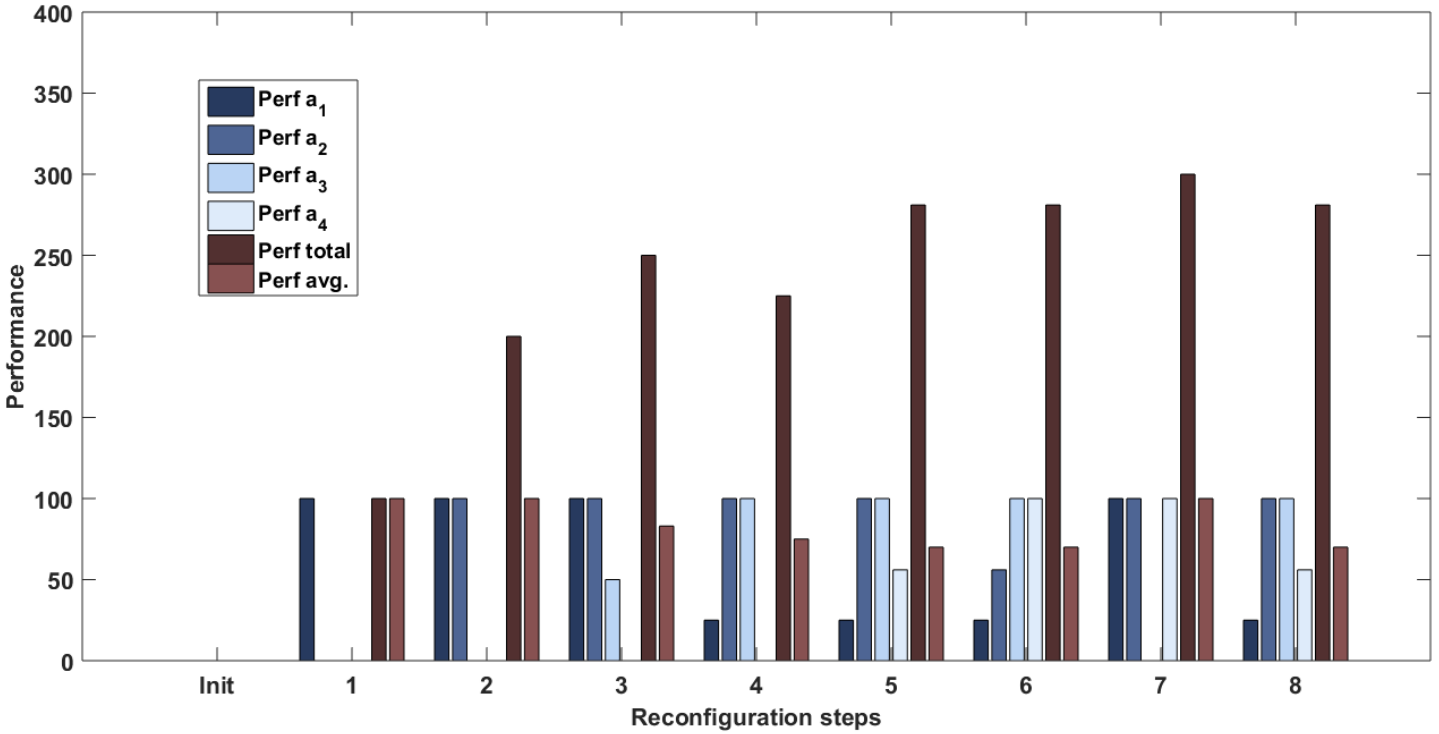
- **Reconfiguration requests**

| **step** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| **request** | $a_1 \rightarrow$ on | $a_2 \rightarrow$ on | $a_3 \rightarrow$ on | $\alpha_3 \rightarrow 1$ | $a_4 \rightarrow$ on | $\alpha_4 \rightarrow 1$ | $a_3 \rightarrow$ off | $a_3 \rightarrow$ on |

- **Implementation**
  - EB6120 [11] as ECUs
  - Software developed with Simulink and SIMTOOLS/SIMTARGET [12]
  - Middleware implemented with Simulink
  - Configuration calculation mapped on a dedicated ECU

# Experimental Results

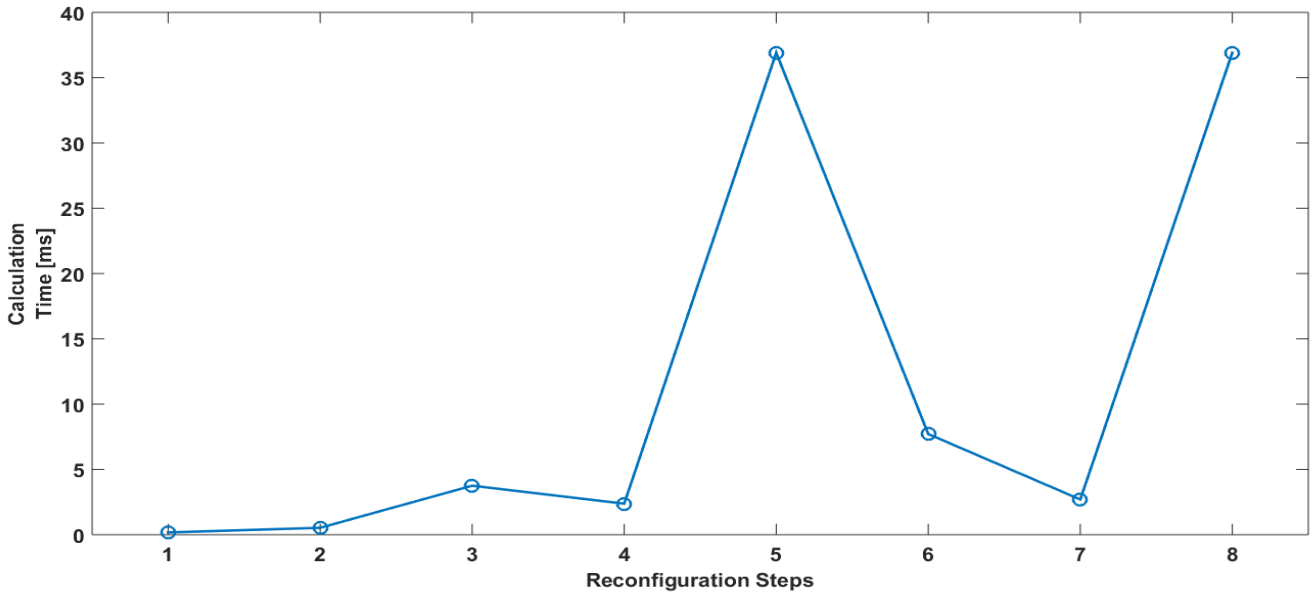
- **Re-configuration according to request sequence**
  - Application and overall performance




| step | request |
|---|---|
| 0 | |
| 1 | $a_1 \rightarrow \text{on}$ |
| 2 | $a_2 \rightarrow \text{on}$ |
| 3 | $a_3 \rightarrow \text{on}$ |
| 4 | $\alpha_3 \rightarrow 1$ |
| 5 | $a_4 \rightarrow \text{on}$ |
| 6 | $\alpha_4 \rightarrow 1$ |
| 7 | $a_3 \rightarrow \text{off}$ |
| 8 | $a_3 \rightarrow \text{on}$ |

# Experimental Results

- **Online configuration synthesis time**
  - EB6120 as ECU
  - Software implementation with Simulink Matlab-embedded functions




| step | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| No. apps | 1 | 2 | 3 | 3* | 4 | 4* | 3 | 4 |

**one application has a fixed mode*

# Concluding Remarks

- **Problem**
  - Online communication reconfiguration of FlexRay-based ECU network
  - Communication resource re-allocation for multi-mode applications and newly activated applications

# Concluding Remarks

- **Problem**
  - Online communication reconfiguration of FlexRay-based ECU network
  - Communication resource re-allocation for multi-mode applications and newly activated applications

- **Approach**
  - A middleware layer with
    - Reconfigurable data-to-schedule mapping
    - Online configuration calculation and deployment
  - Achieves online re-allocation of communication resources with an ECU

# Concluding Remarks

- **Problem**
  - Online communication reconfiguration of FlexRay-based ECU network
  - Communication resource re-allocation for multi-mode applications and newly activated applications

- **Approach**
  - A middleware layer with
    - Reconfigurable data-to-schedule mapping
    - Online configuration calculation and deployment
  - Achieves online re-allocation of communication resources with an ECU

- **Outlook**
  - Extension to support other communication protocols
  - Extension to allow features like re-routing of messages
  - More efficient configuration calculation algorithms

# References


[1] P. Mundhenk, F. Sagstetter, S. Steinhorst, M. Lukasiewycz, and S. Chakraborty. Policy-based message scheduling using FlexRay. CODES+ISSS, 2014.

[2] L. T. Phan, I. Lee, and O. Sokolsky. A semantic framework for mode change protocols. RTAS, 2011.

[3] L. Sha, R. Rajkumar, J. Lehoczky, and K. Ramamritham. Mode change protocols for priority-driven preemptive scheduling. Real-Time Systems, 1989

[4] J. Real, and A. Crespo. Mode change protocols for real-time systems: a survey and a new proposal. Real-time Systems, 2004.

[5] G. Fohler. Changing operational modes in the context of pre run-time scheduling. IEICE Transactions on Information and Systems, 1993.

[6] K. Klobedanz, A. Koenig, and W. Mueller. A reconfiguration approach for faulte-tolerant FlexRay networks. DATE, 2011.

[7] FlexRay communications system protocol specification, Version 2.1. www.flexray.com. 2005.

[8] M. Lukasiewycz, M. Glaß, J. Teich, and P. Milbredt. FlexRay schedule optimization of the static segment. CODES+ISSS, 2011.

[9] R. Schneider, D. Goswami, S. Zafar, M. Lukasiewycz, and S. Chakraborty. Constraint-driven synthesis and tool-support for FlexRay-based automotive control systems. CODES+IEEE, 2011.

[10] D. Goswami, R. Schneider, and S. Chakraborty. Relaxing signal delay constraints in distributed embedded controllers. IEEE Trans. Contr. Sys. Techn., 2014

[11] Elektrobit. www.elektrobit.com.

[12] SIMTOOLS/SIMTARGET. www.simtools.at

# The End

**Many thanks**

**Q/A**