

# On the Performance of UML State Machine Interpretation at Runtime

*Edzard Höfig, Peter H. Deussen, Ina Schieferdecker*



# Motivation



## Autonomic Systems Engineering

### **Self-Awareness\***

The ability of a system to perceive its own operational state

▣▣▣▣ Requires representation of system behaviour

▣▣▣▣ Use of behaviour models + runtime interpretation

### **Homeostasis\*\***

The ability of a system to adapt to environmental changes

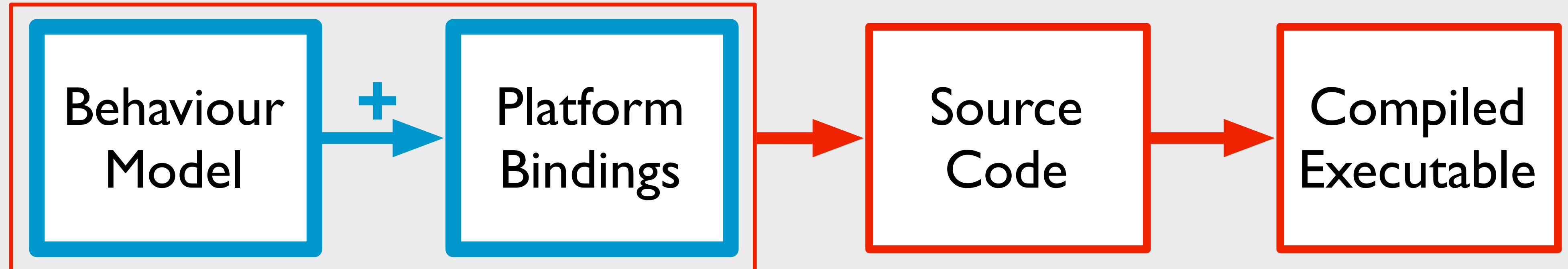
▣▣▣▣ Requires support for functional scalability at runtime

\* See M. Smirnow et al., *Demystifying Self-awareness of Autonomic Systems*, 2009 2 of 12

\*\* See Chapter 5 in W. R. Ashby, *Design for a Brain*, Chapman & Hall, 1960

# Execution of Behaviour Models

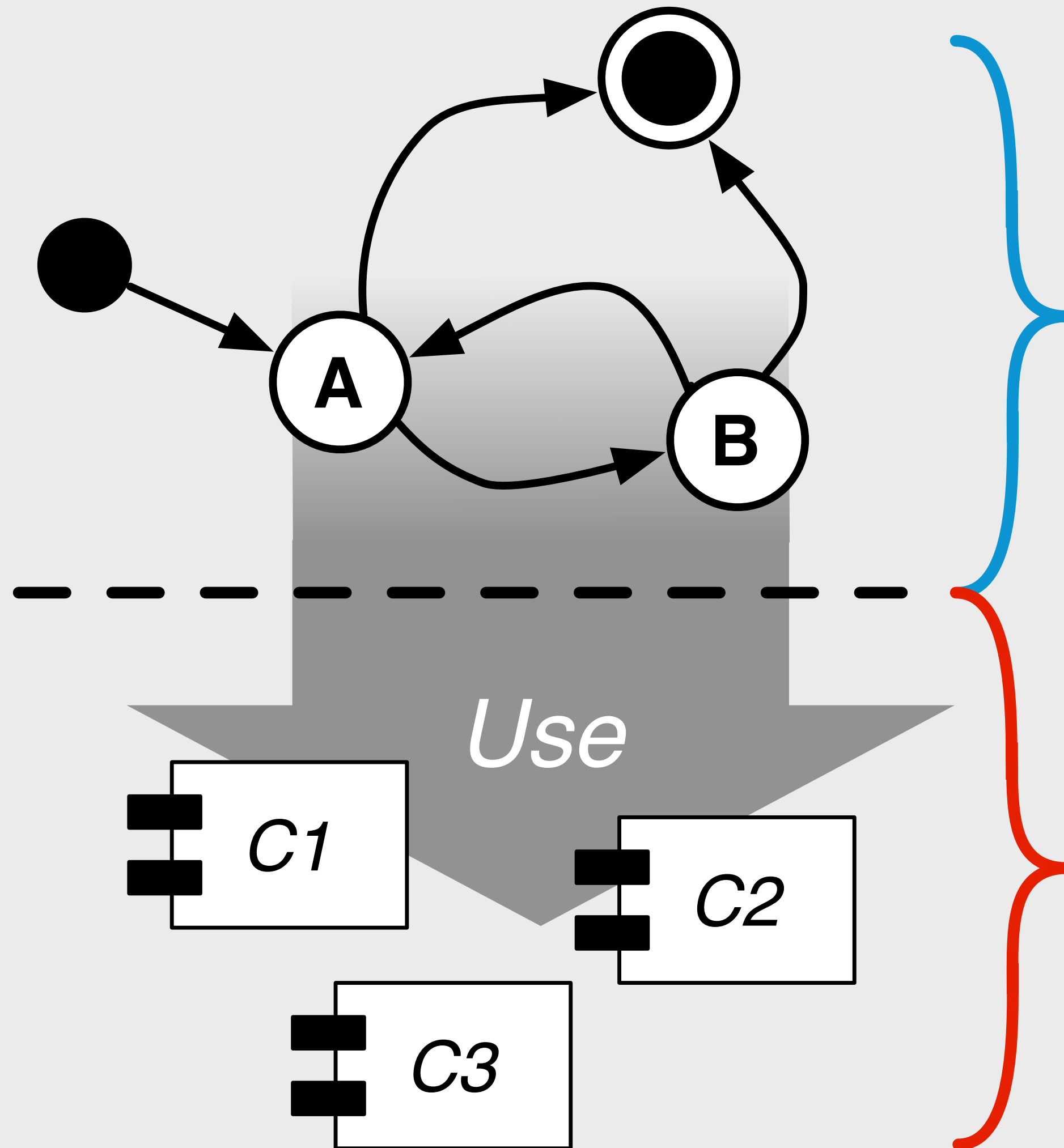
**Interpretation** is different than **compilation**



➡ This approach makes runtime adaptation easy

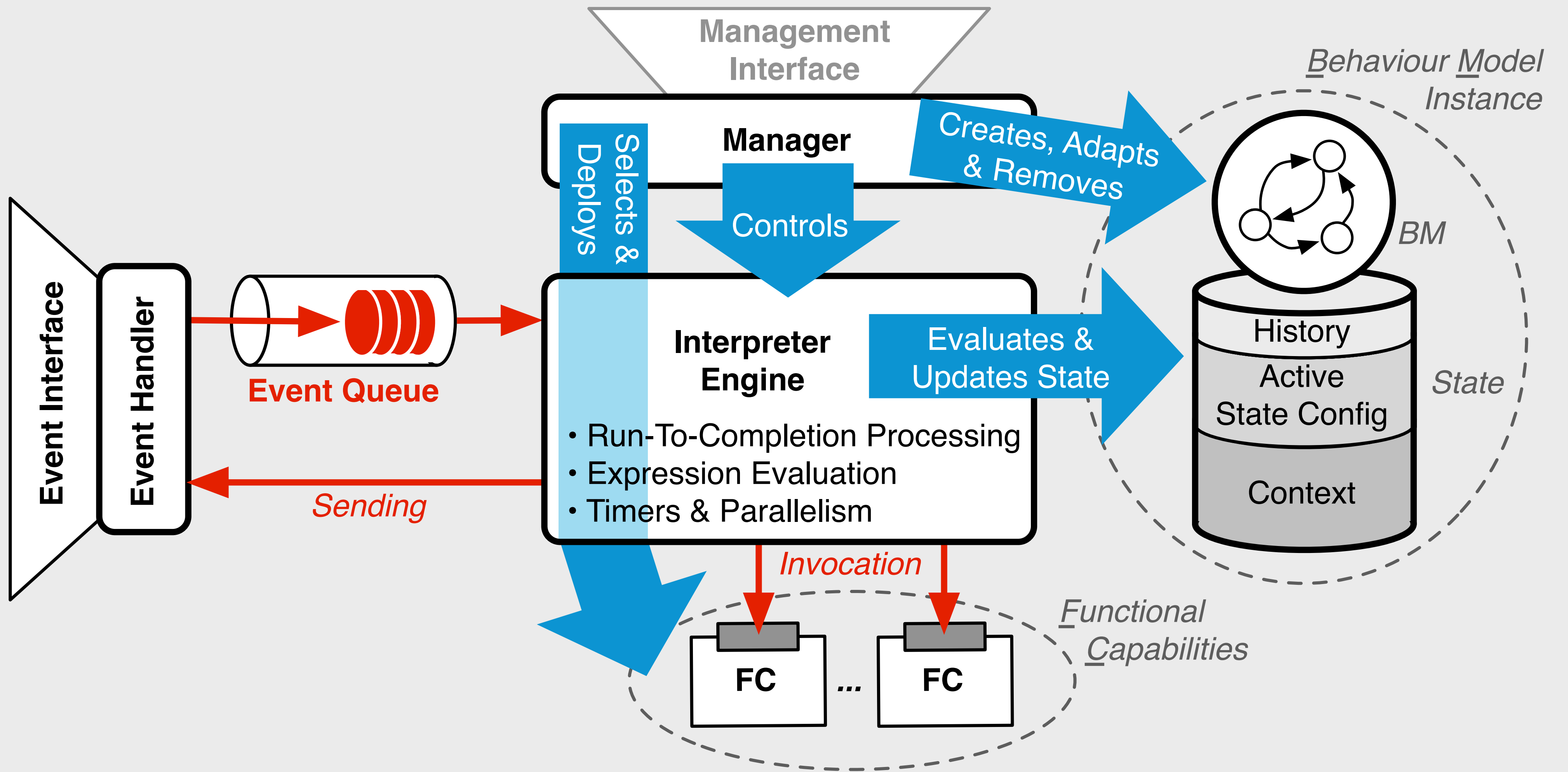
➡ With this approach runtime adaptation is difficult or impossible

# Splitting Behaviour Representation



**Application logic as model  
(UML State Machines) —  
interpreted at runtime**

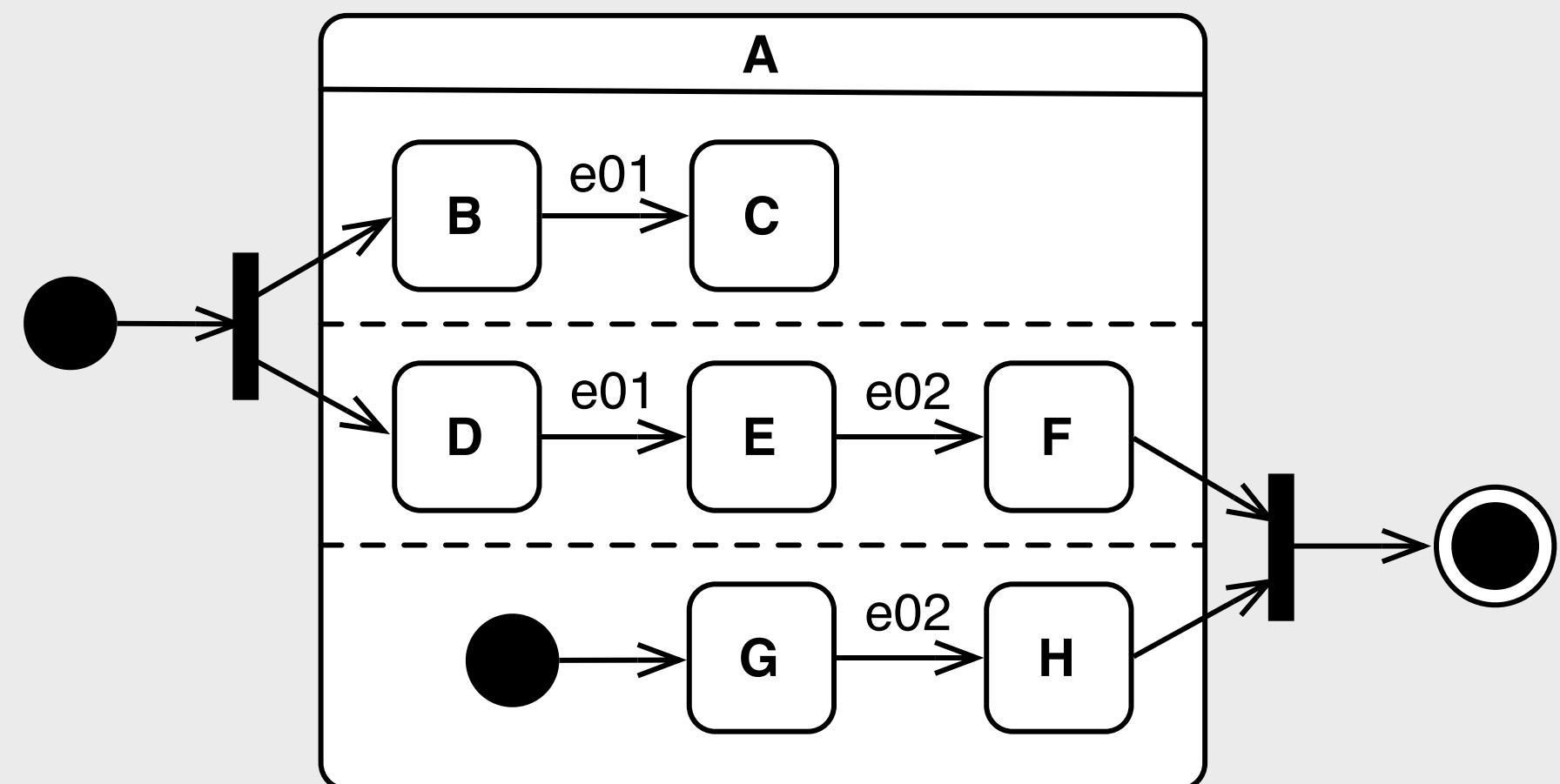
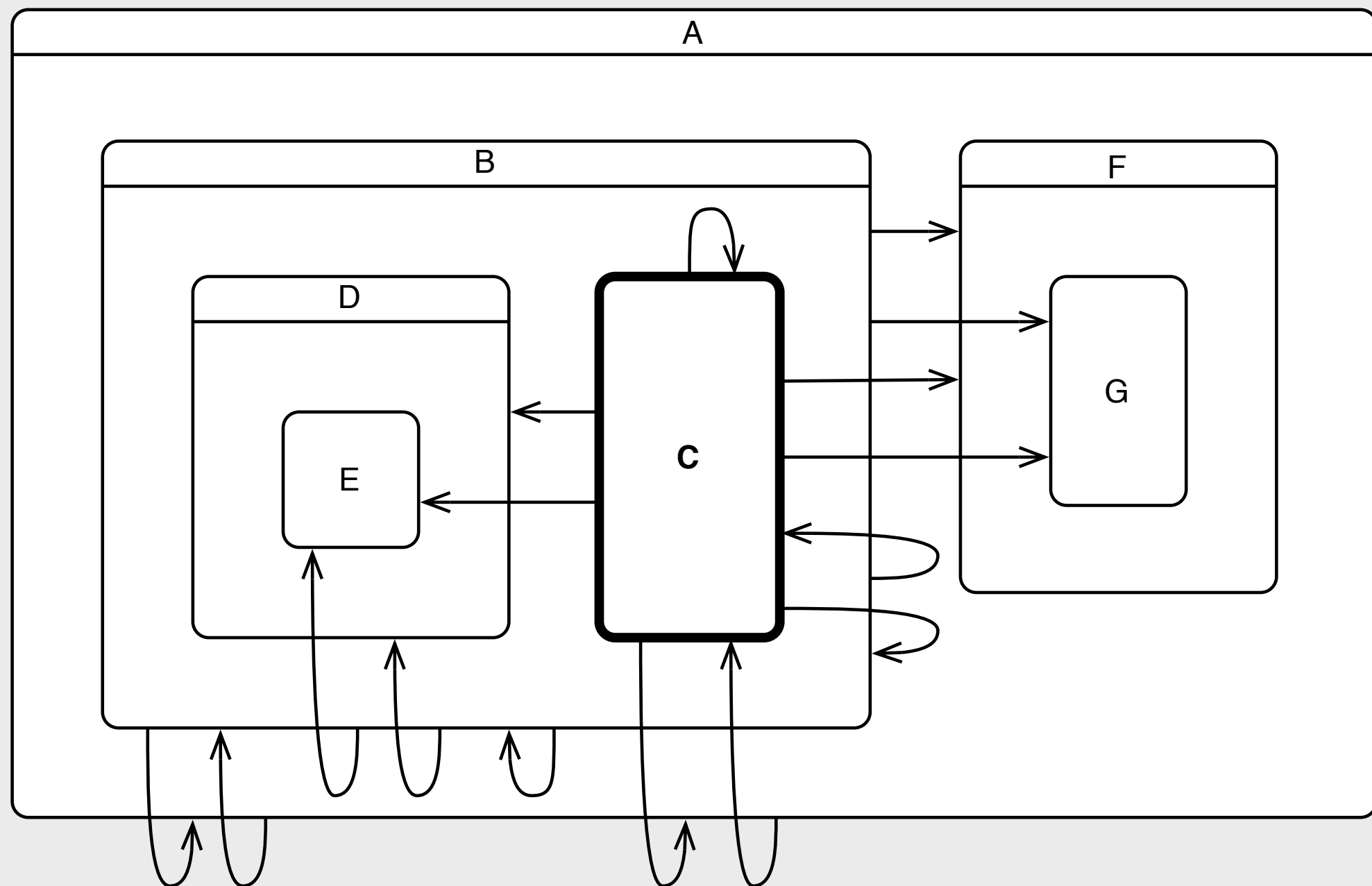
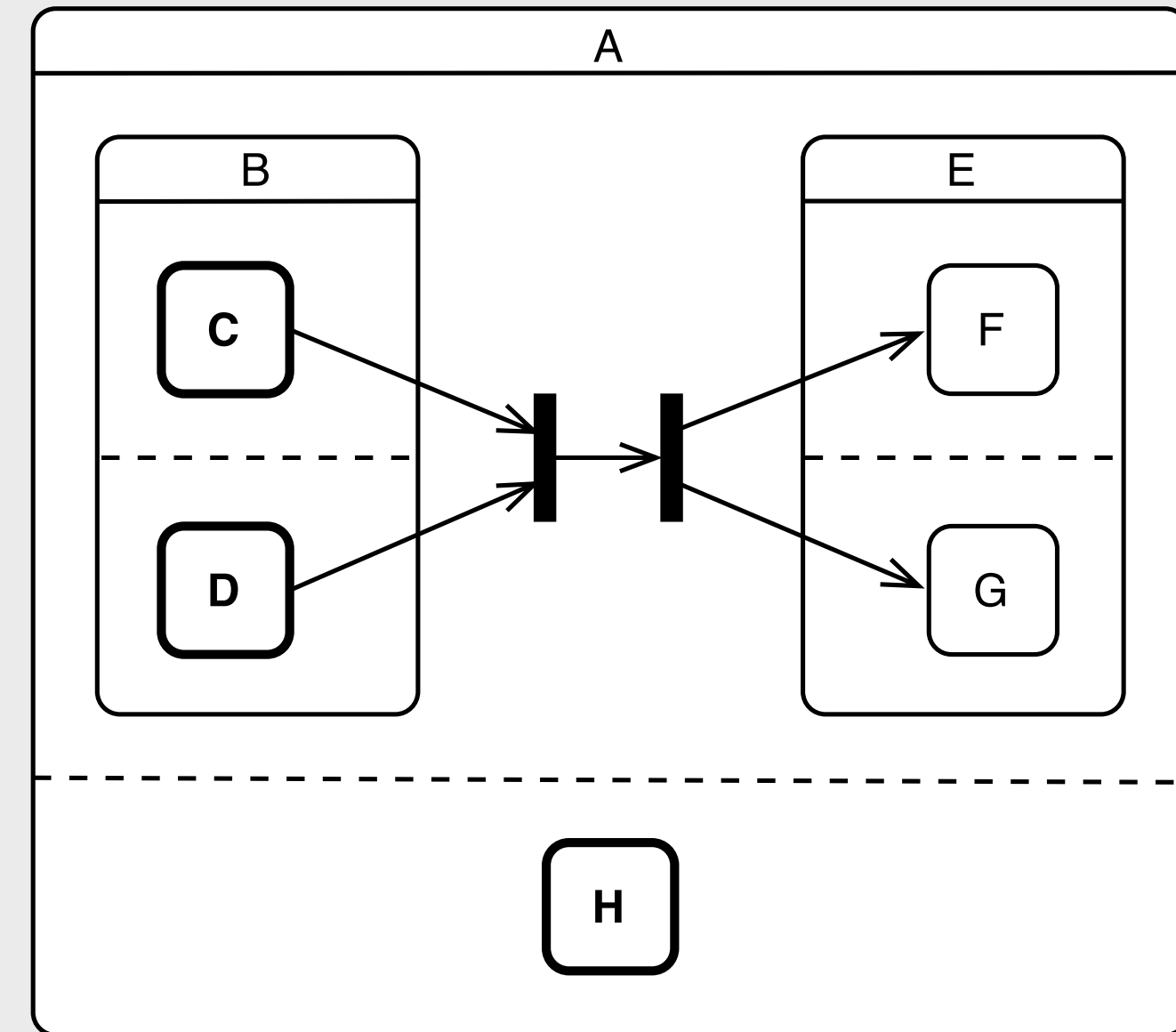
**Platform code as static  
components — dynamic  
deployment & linkage**



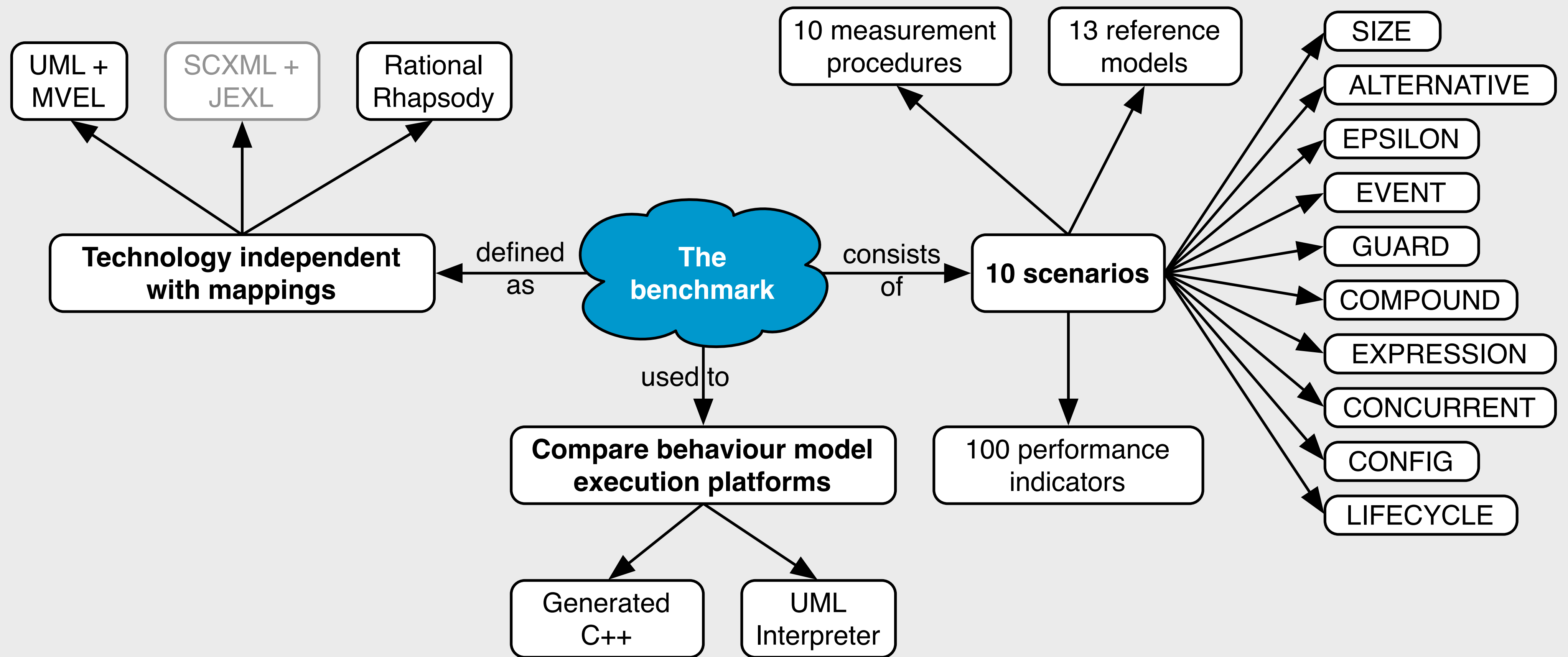
# Conceptual Interpreter Architecture

# Some Challenges...

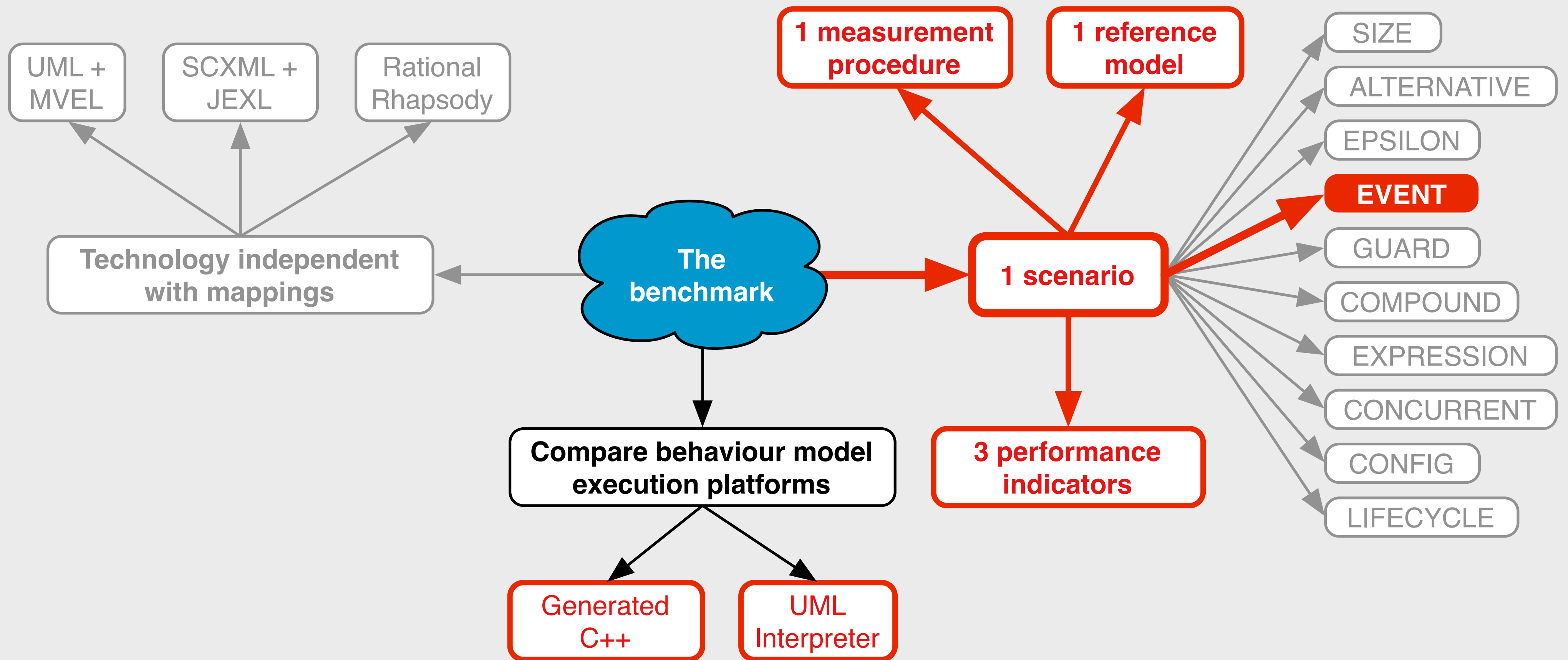
- Processing compound transitions
- Determination of entered/exited states
- Implicit Fork / Join



# Performance Benchmark

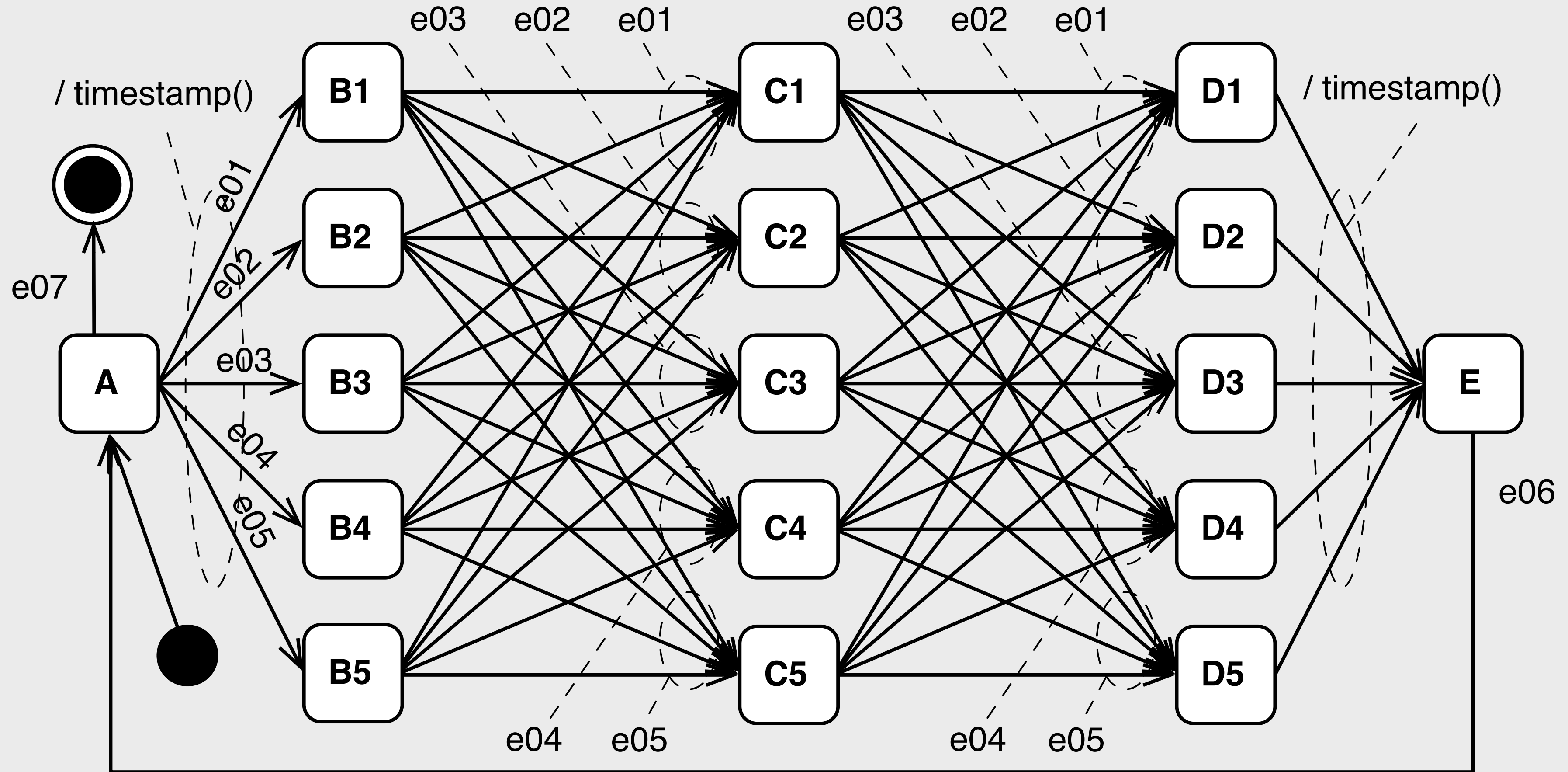


# Performance Benchmark



# Behaviour Model

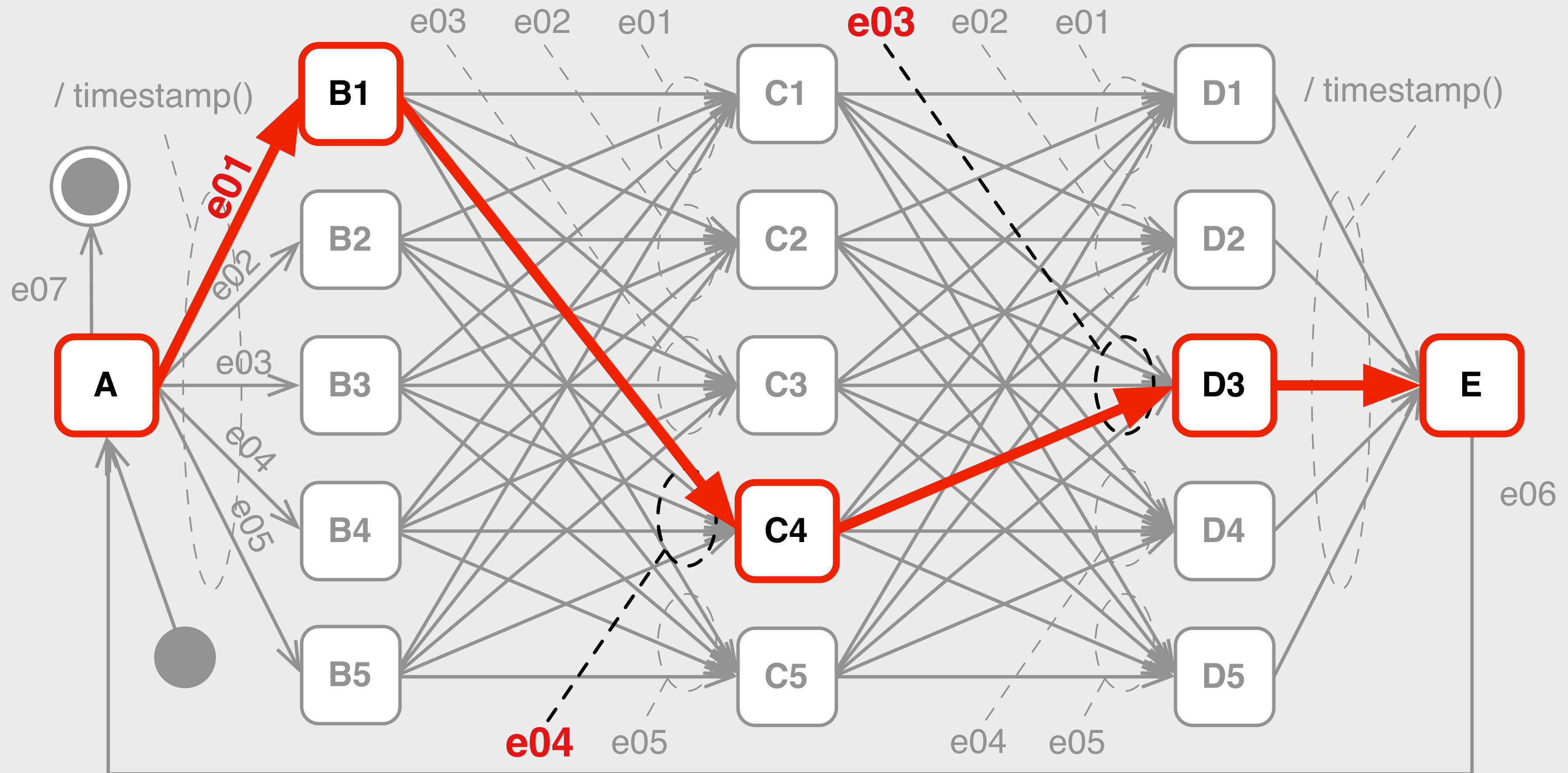
## EVENT Scenario



# Behaviour Model

## EVENT Scenario

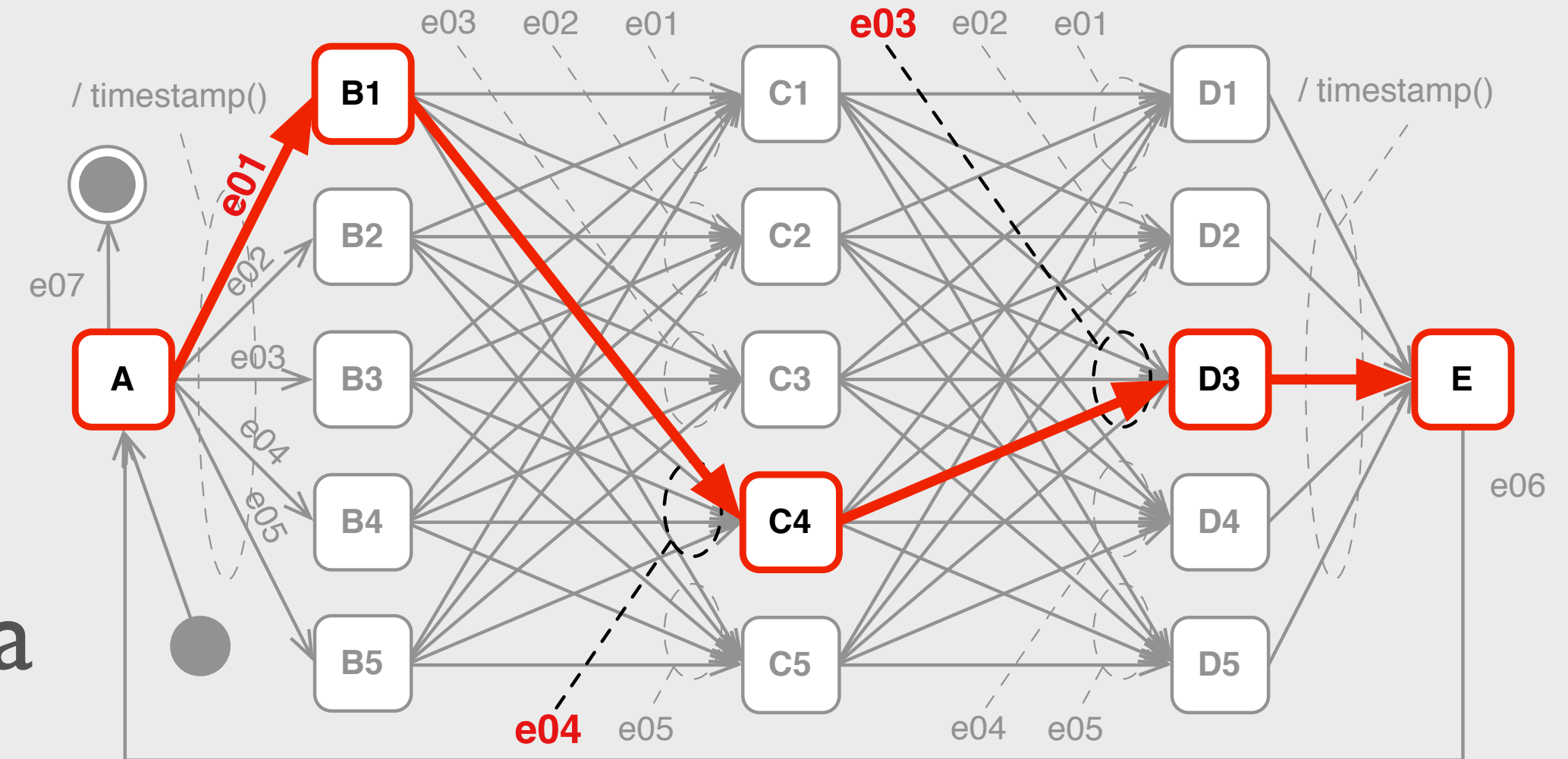
$5^3 = 125$  different paths



# Measurement Procedure

## EVENT Scenario

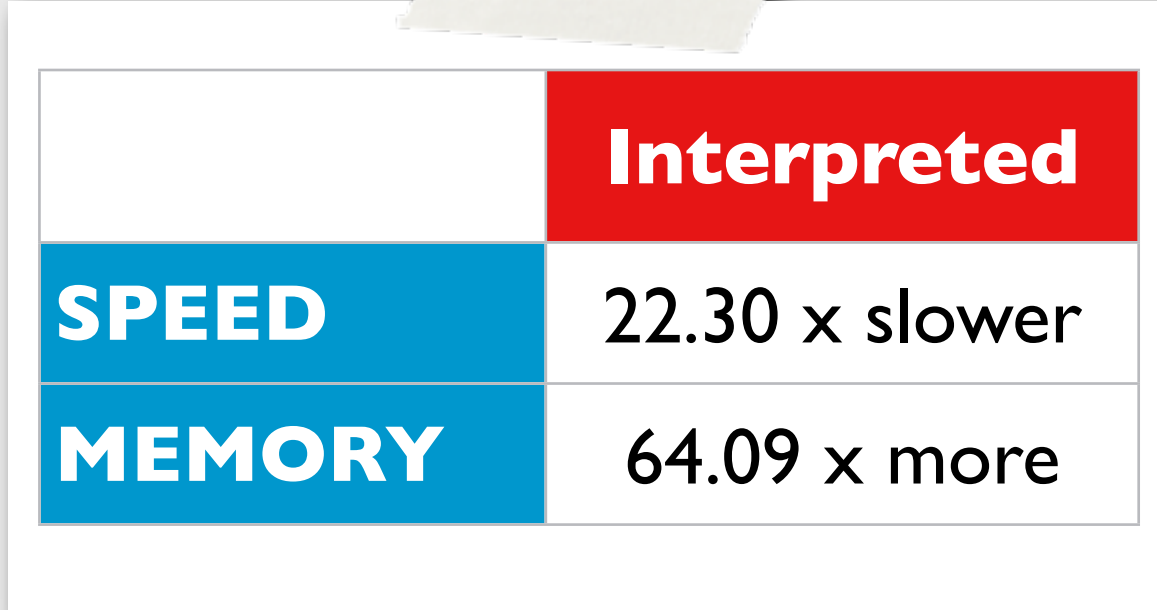
- Execute all 125 paths (a „trial“)
  - ▣▣▣▣▣ Uses all event combinations
- Repeat trials 125 times, starting with a different path each time
  - ▣▣▣▣▣ Averages measurements
  - ▣▣▣▣▣ Distributes build-up effects
- Process 15625 timestamp pairs to calculate three performance indicators



	Compiled	Interpreted
EVENT.MIN	1 $\mu$ s	165 $\mu$ s
EVENT.AVG	23.38 $\mu$ s	186.45 $\mu$ s
EVENT.MAX	128 $\mu$ s	492 $\mu$ s

# Results

- Proposed approach is feasible
- Adequate performance
  - **Exception:** delay-sensitive systems with **timing constraints**  $\ll$  1ms
- Impact of model structure & model storage format



	Interpreted
SPEED	22.30 x slower
MEMORY	64.09 x more

# Benefits for **System Adaptation** and **Self-Management**

- **Consistency** has to be ensured for only a small number of artefacts
  - Artefacts **readily available** during runtime, with **well-defined formats**
- State-transition systems are **reactive**
  - **Quiescent** between processing steps
- Application **state is explicitly** managed
- Models are an **abstraction** of system behaviour
  - **Intuitive** & **formalised**
- **Model checking, simulation, testing**, even **proofs** are possible

# Thanks for Listening!

Questions?

Edzard Höfig

email: [edzard.hoefig@fokus.fraunhofer.de](mailto:edzard.hoefig@fokus.fraunhofer.de)

web: [www.fokus.fraunhofer.de/go/motion](http://www.fokus.fraunhofer.de/go/motion)



**Fraunhofer**  
**FOKUS**

# UML Interpreter + MVEL

→ TIMESTAMP

→ SIZE

→ ALTERNATIVE

→ EPSILON

→ EVENT

→ GUARD

→ COMPOUND

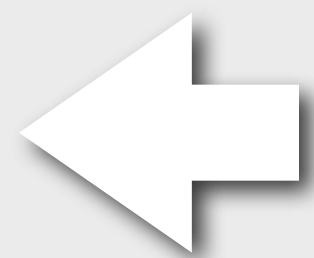
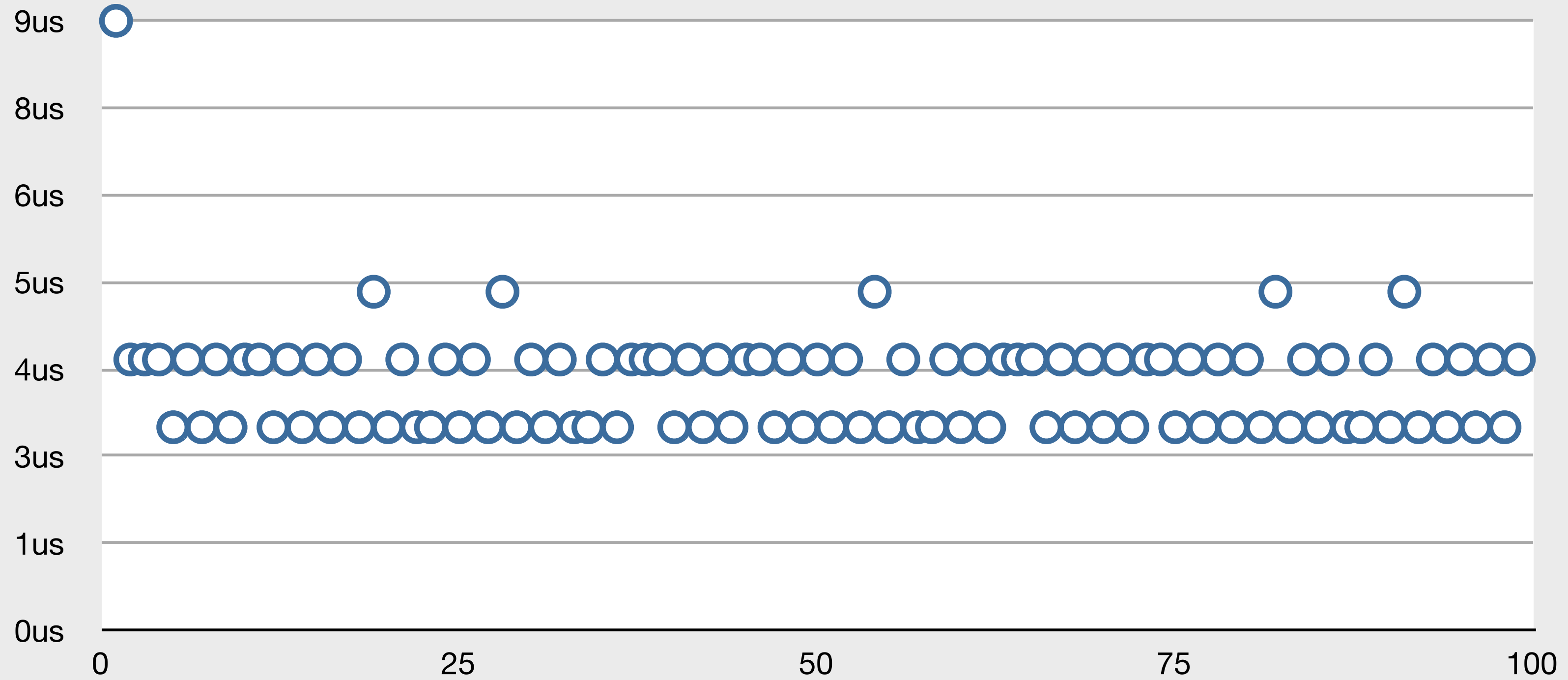
→ EXPRESSION

→ CONCURRENT

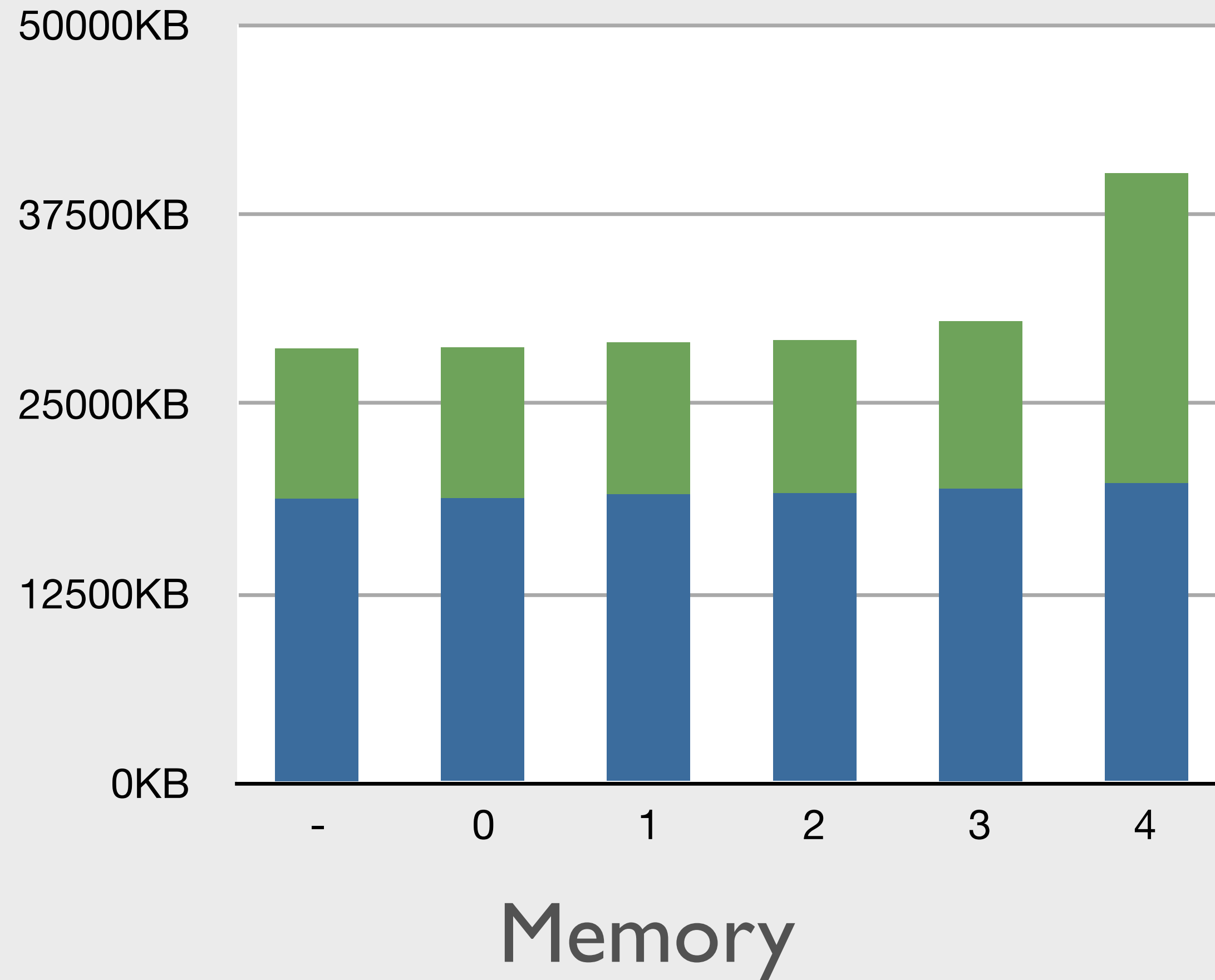
→ CONFIG

→ LIFECYCLE

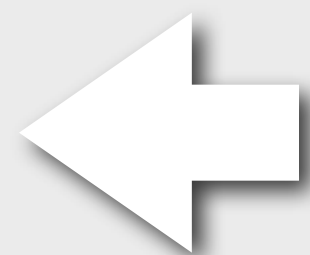
# TIMESTAMP



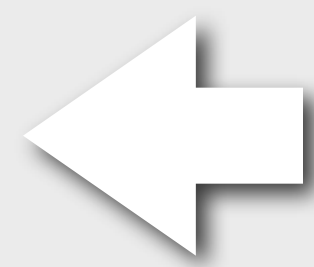
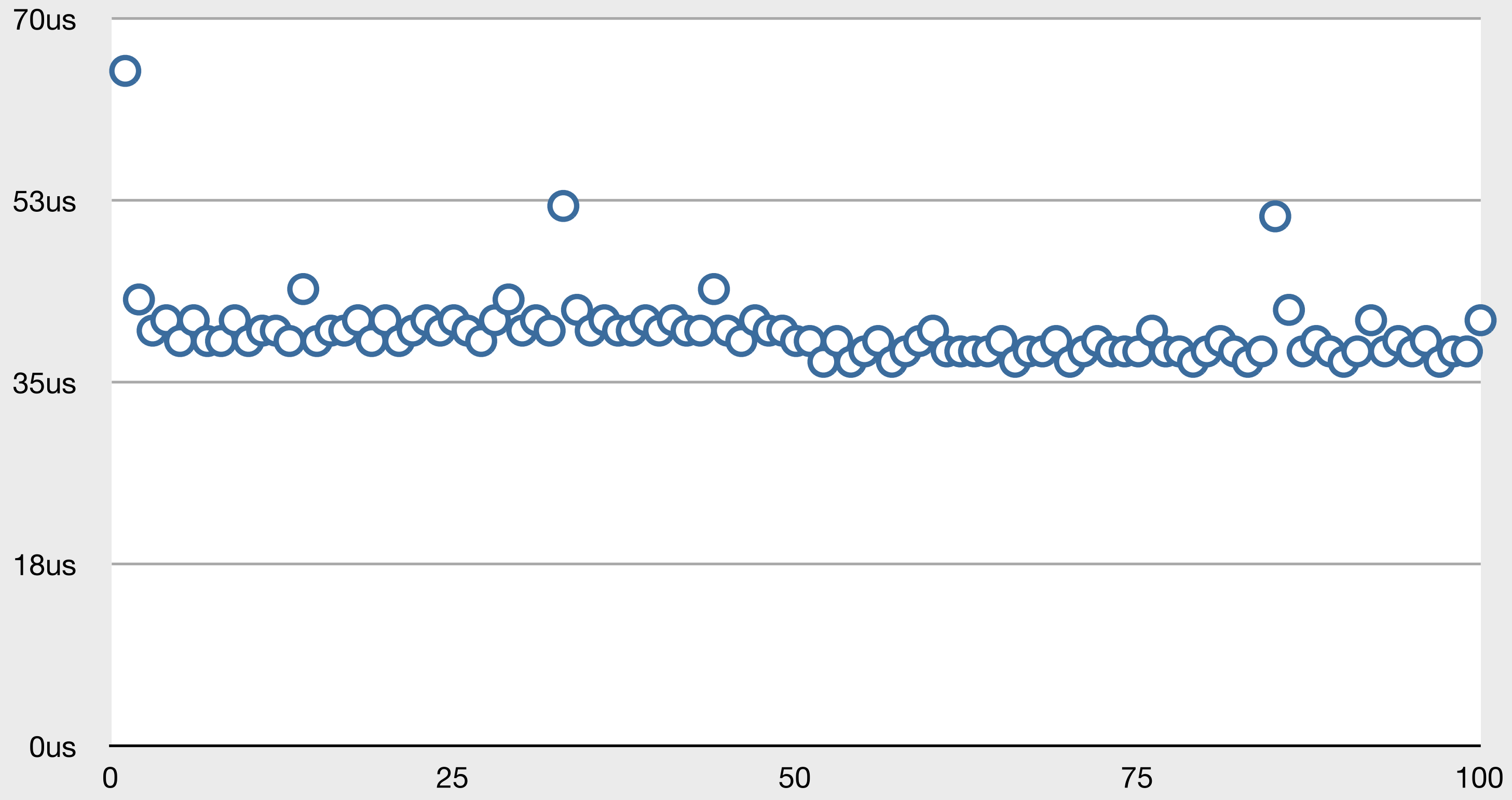
# SIZE



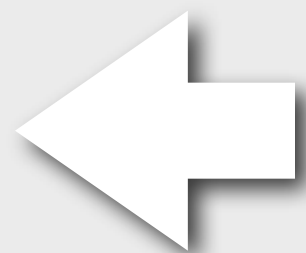
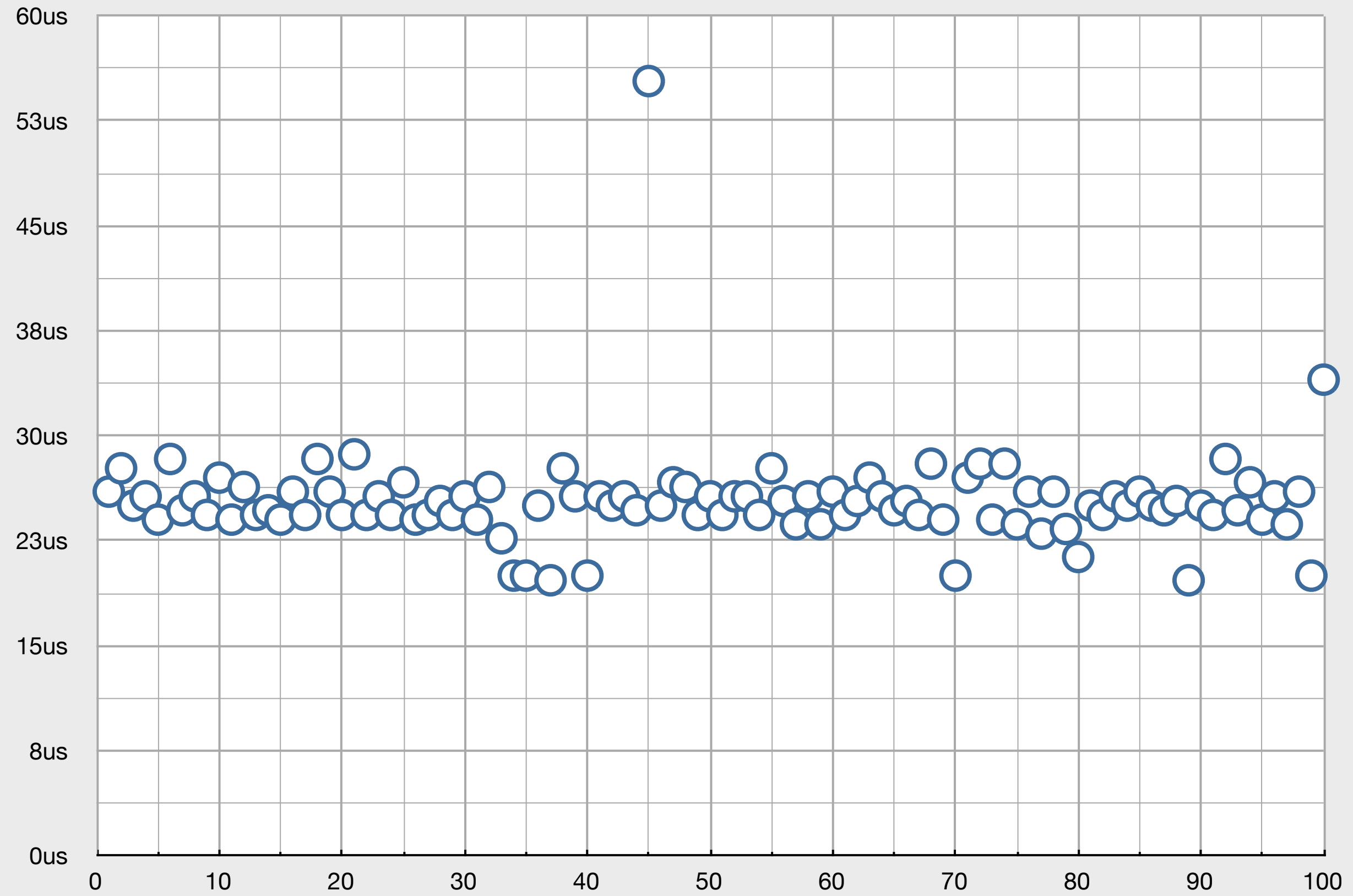
Executable size is always size of JVM

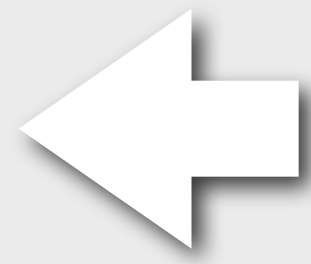


# ALTERNATIVE



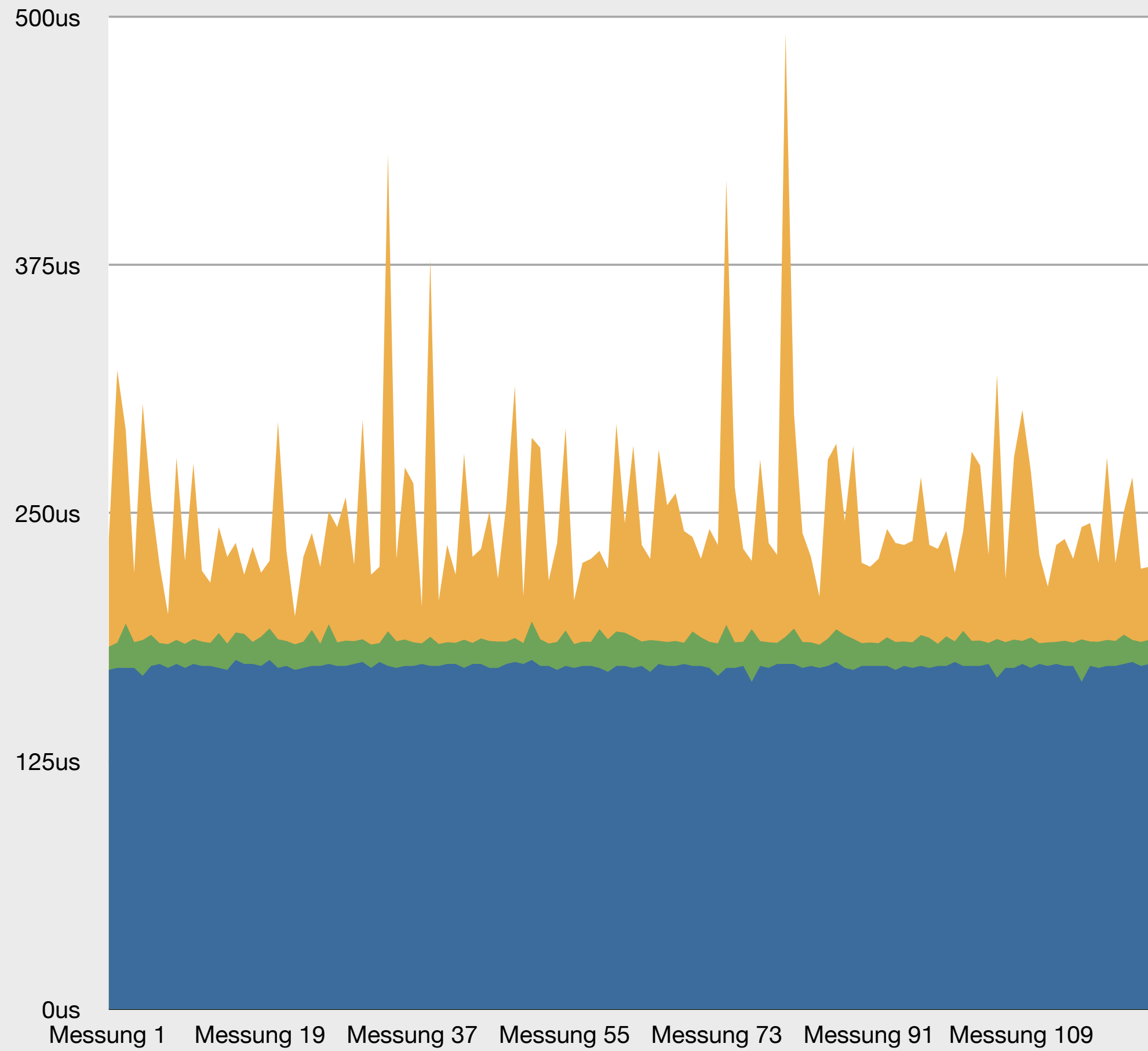
# EPSILON



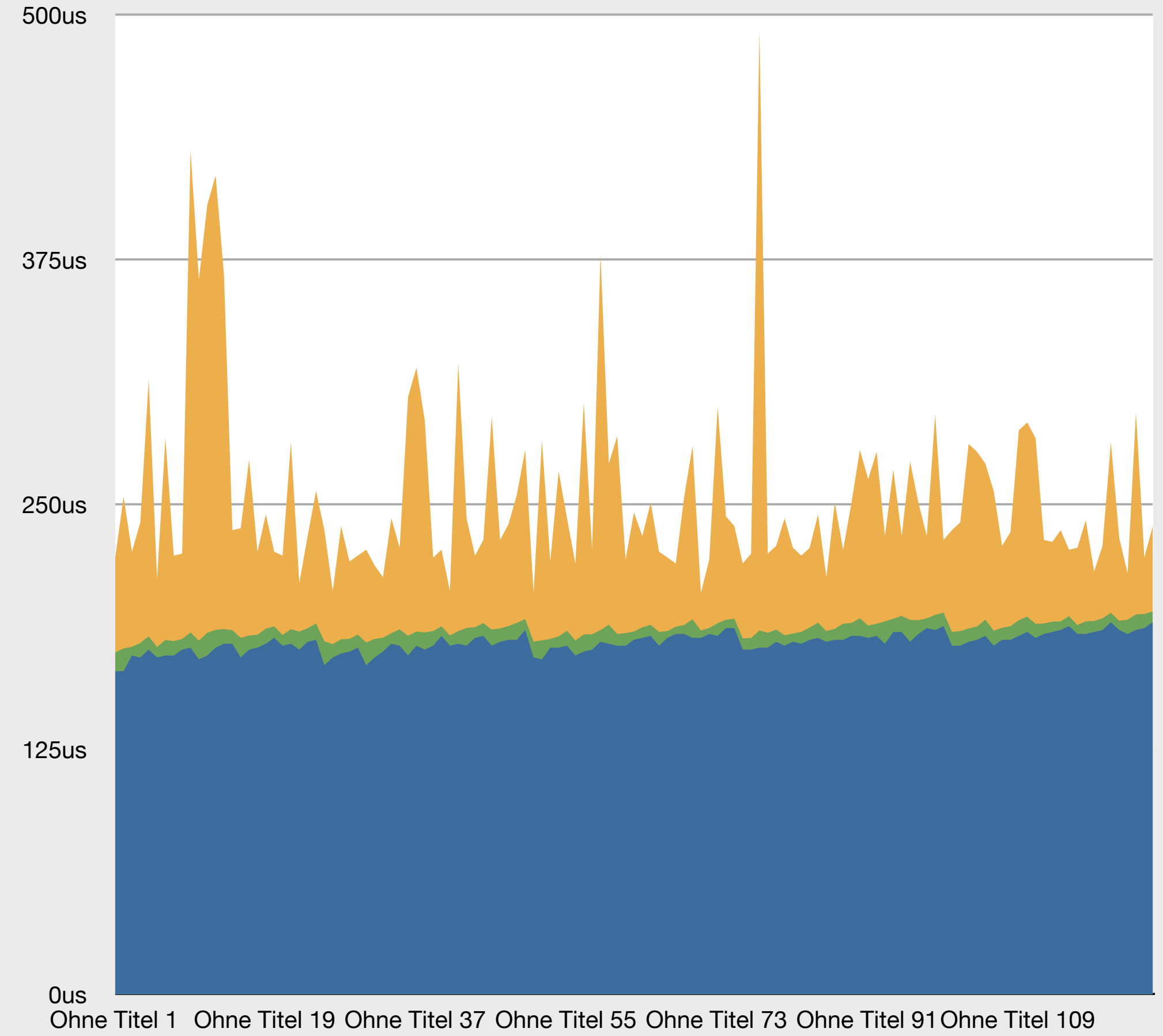


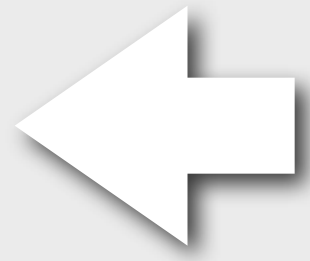
# EVENT

## Trials



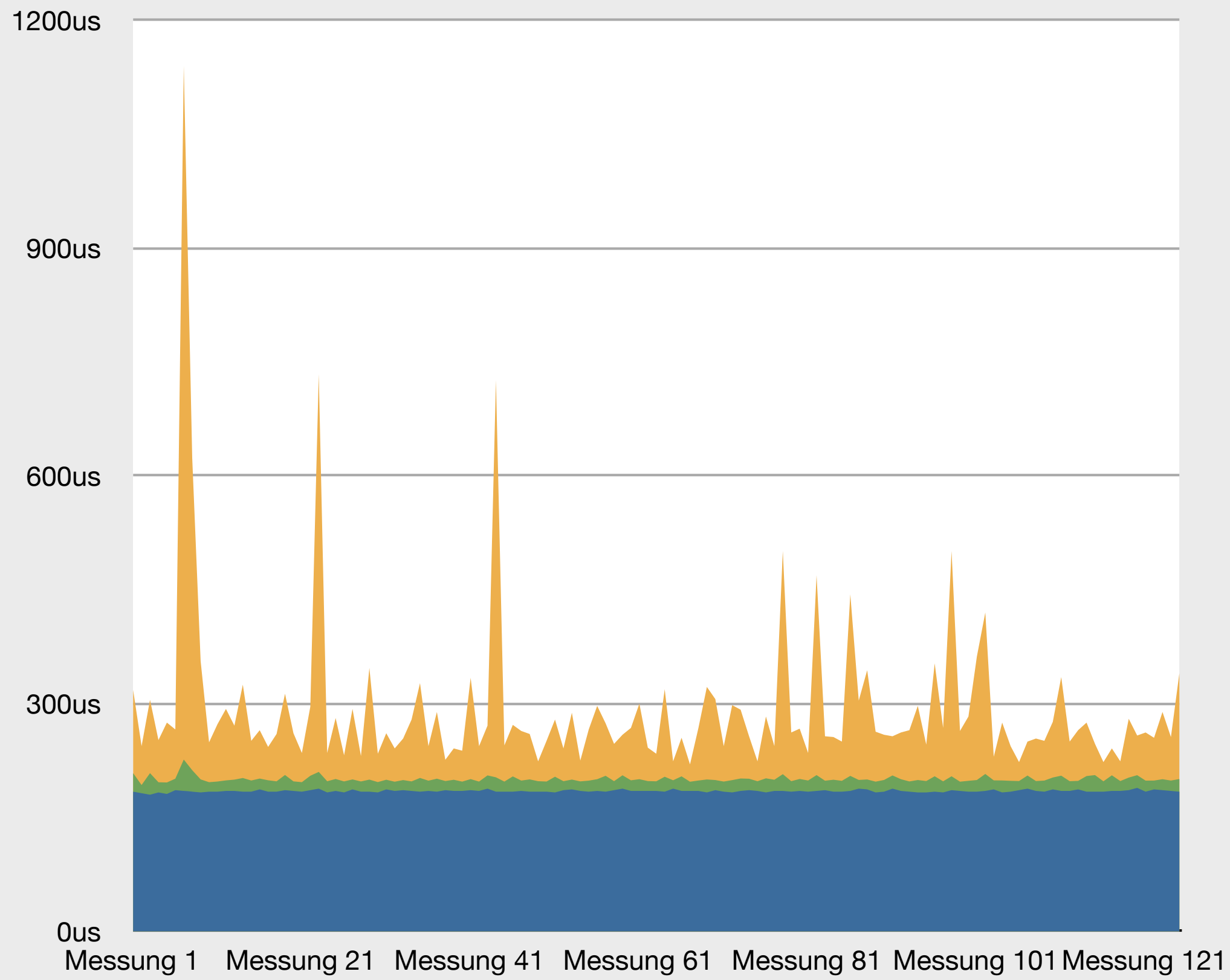
## Paths



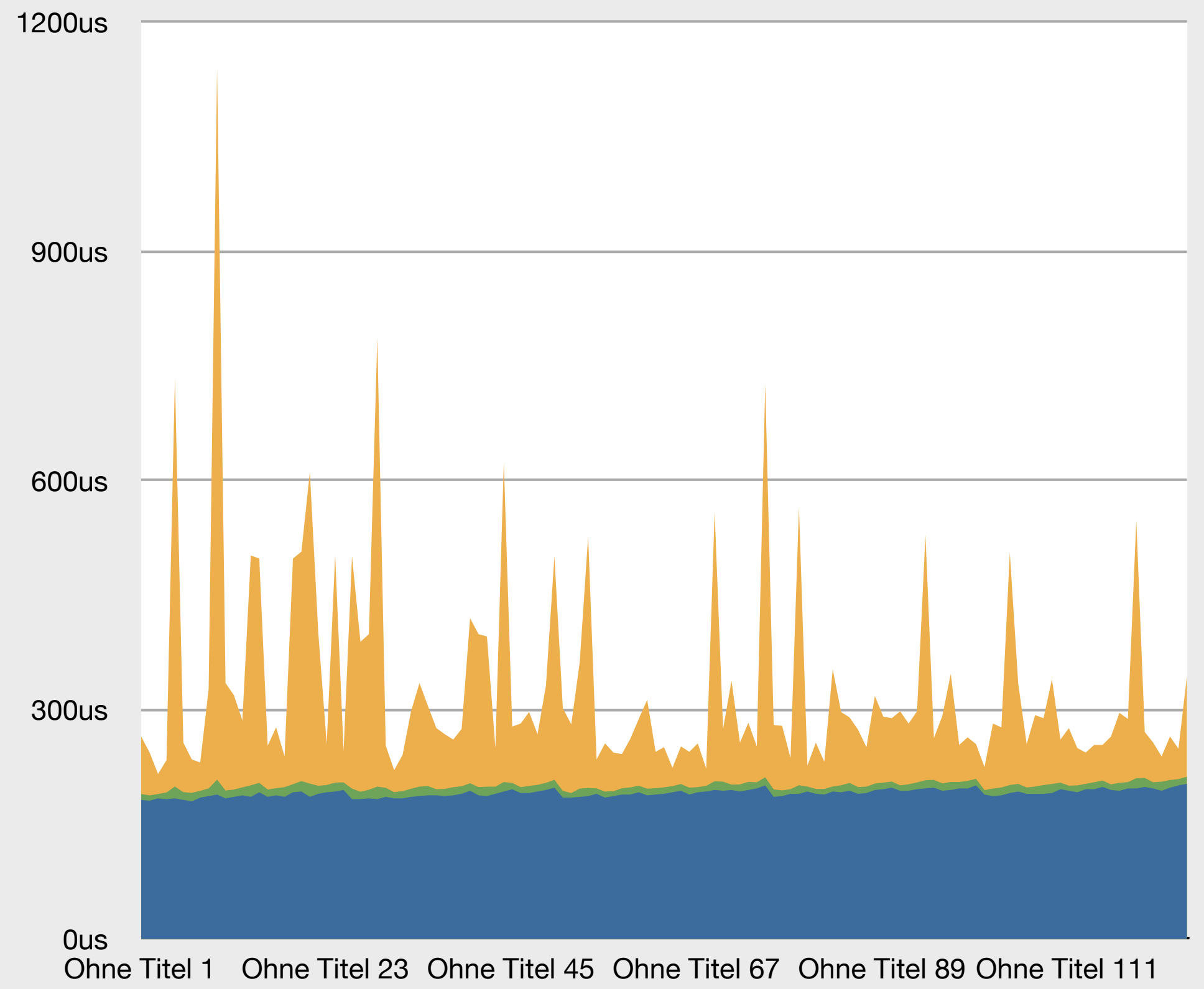


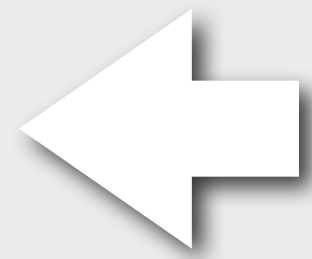
# GUARD

## Trials



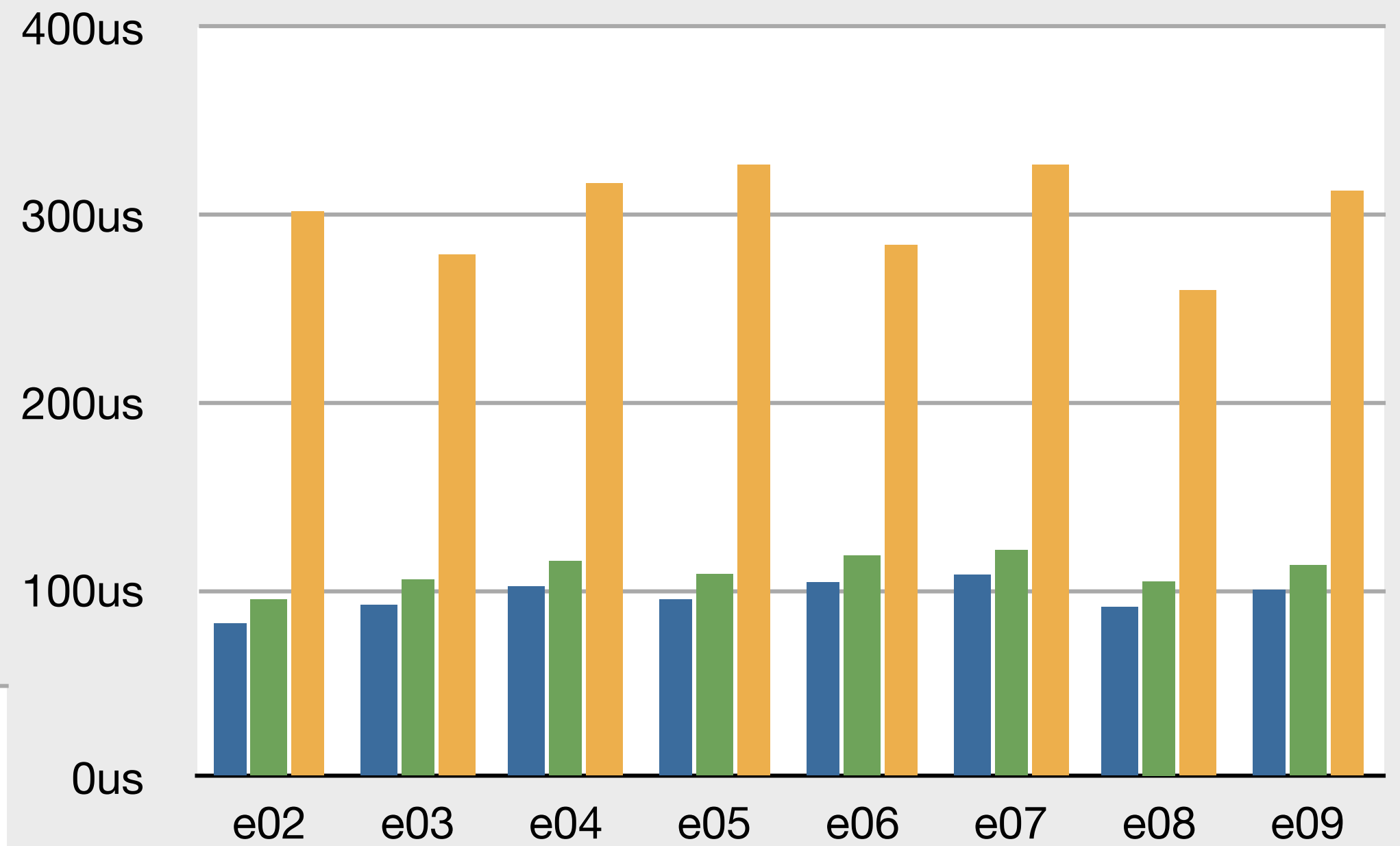
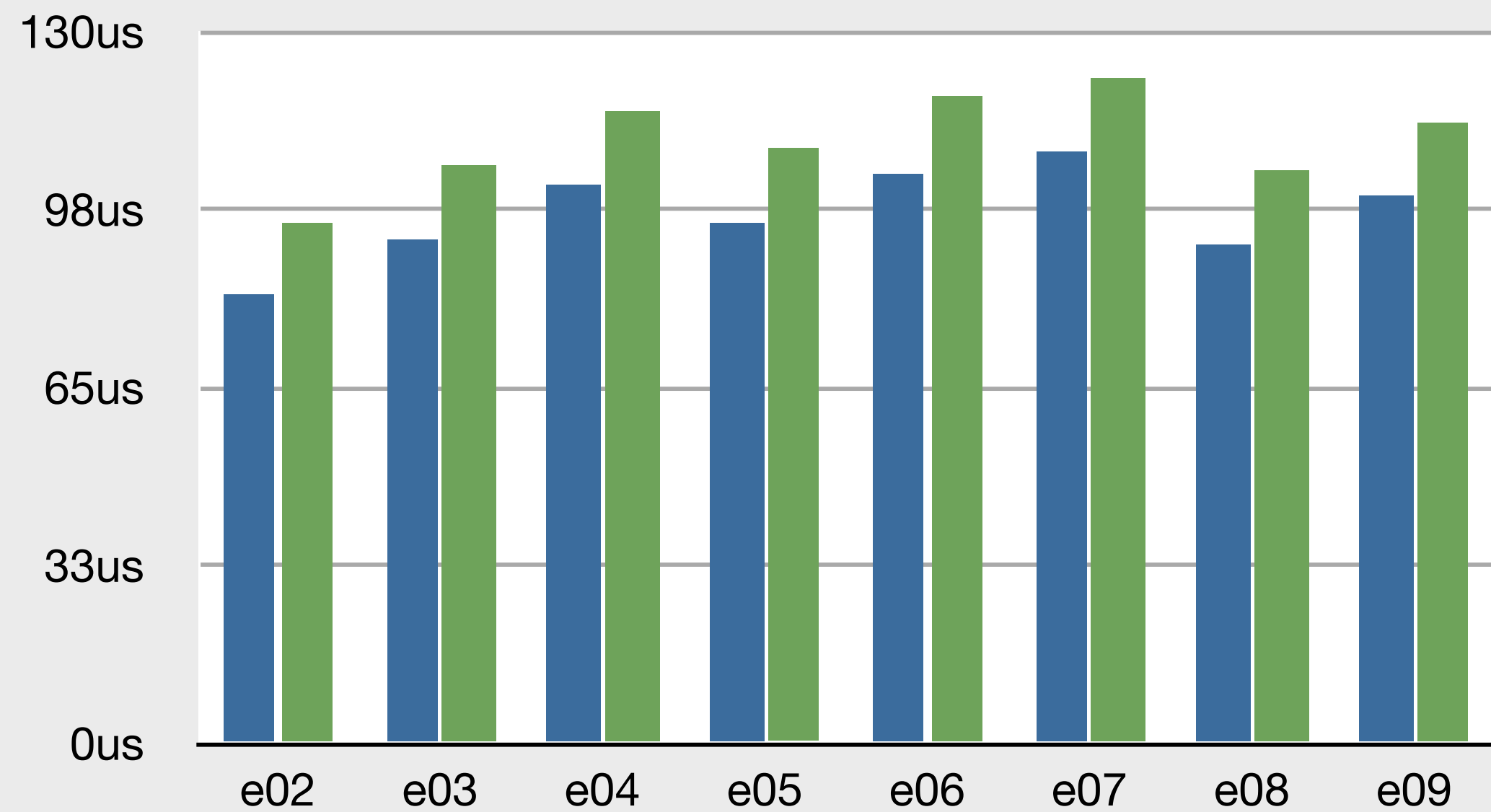
## Paths



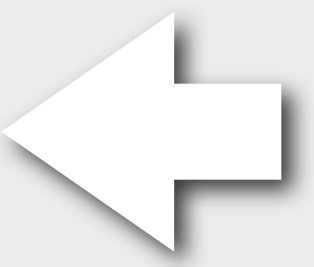
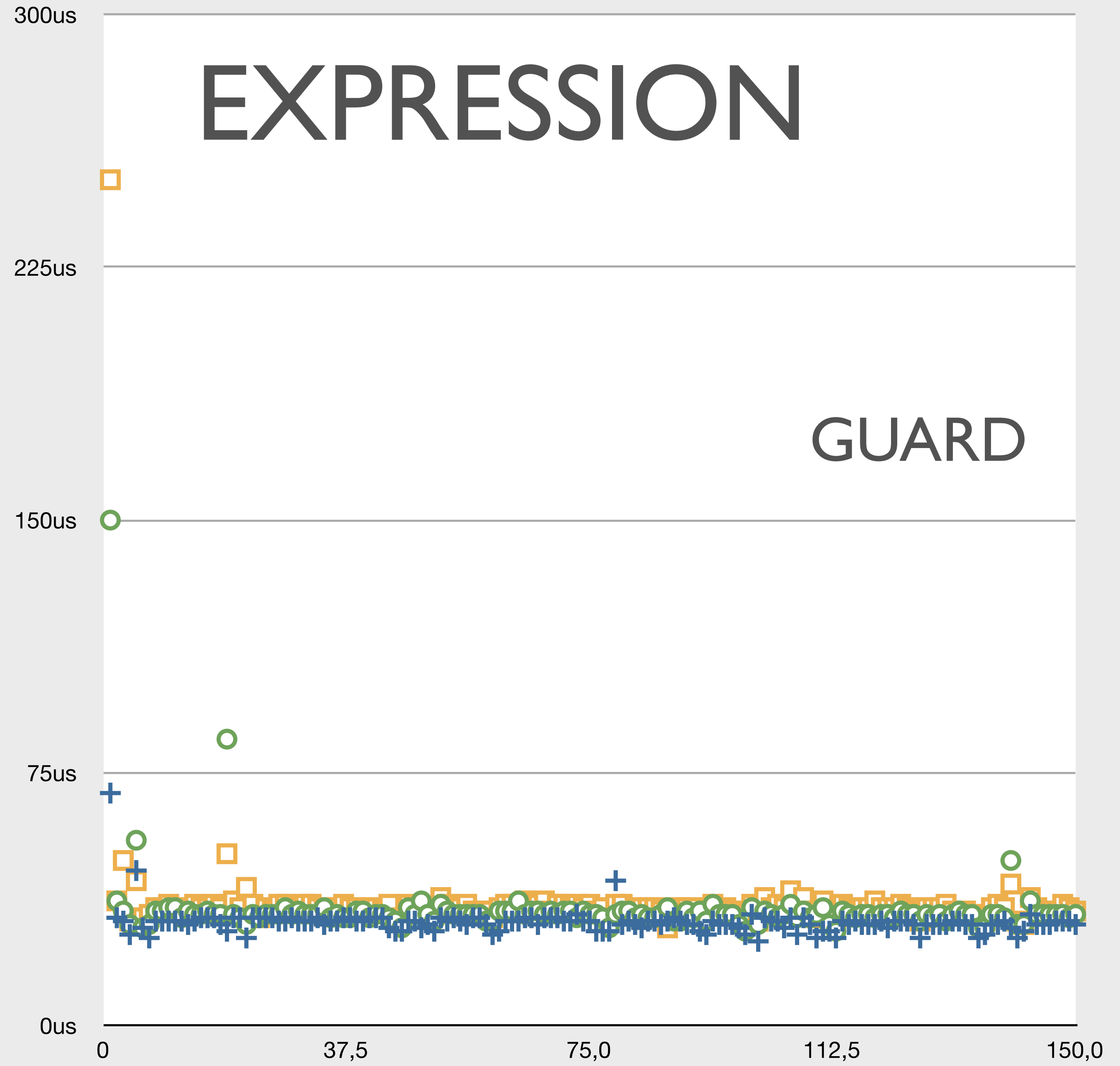
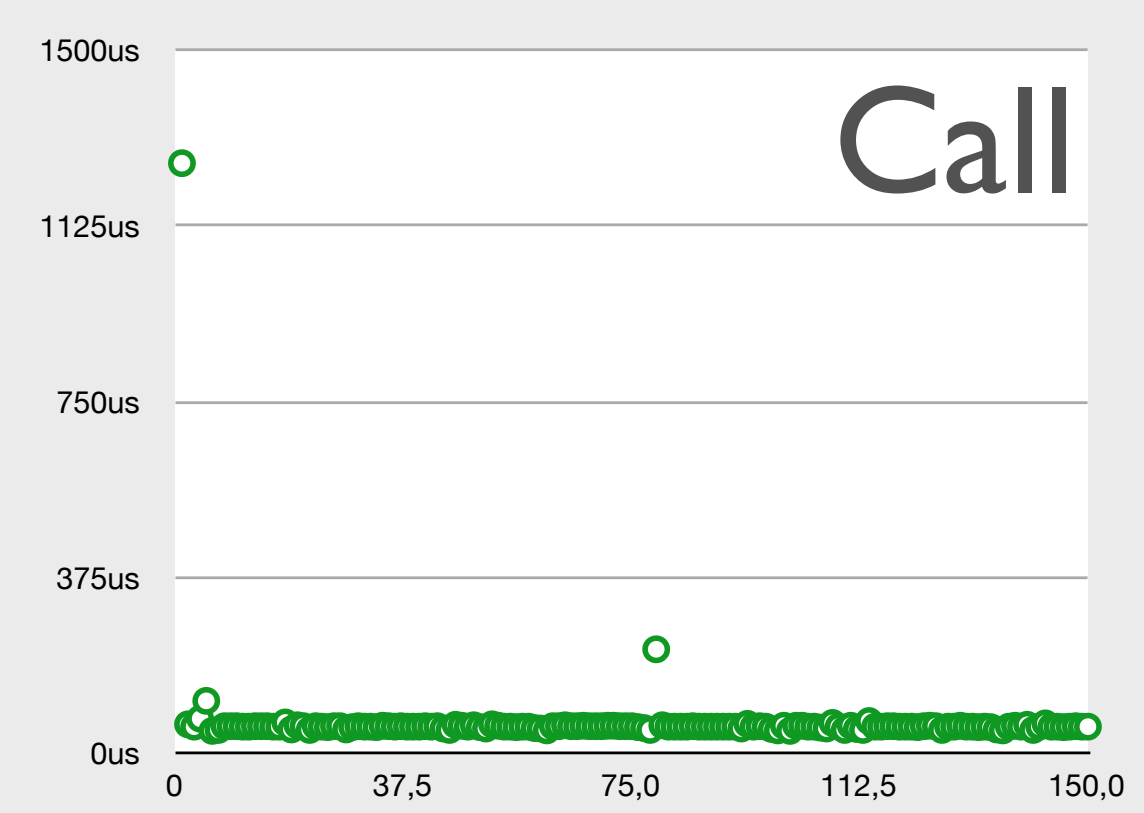
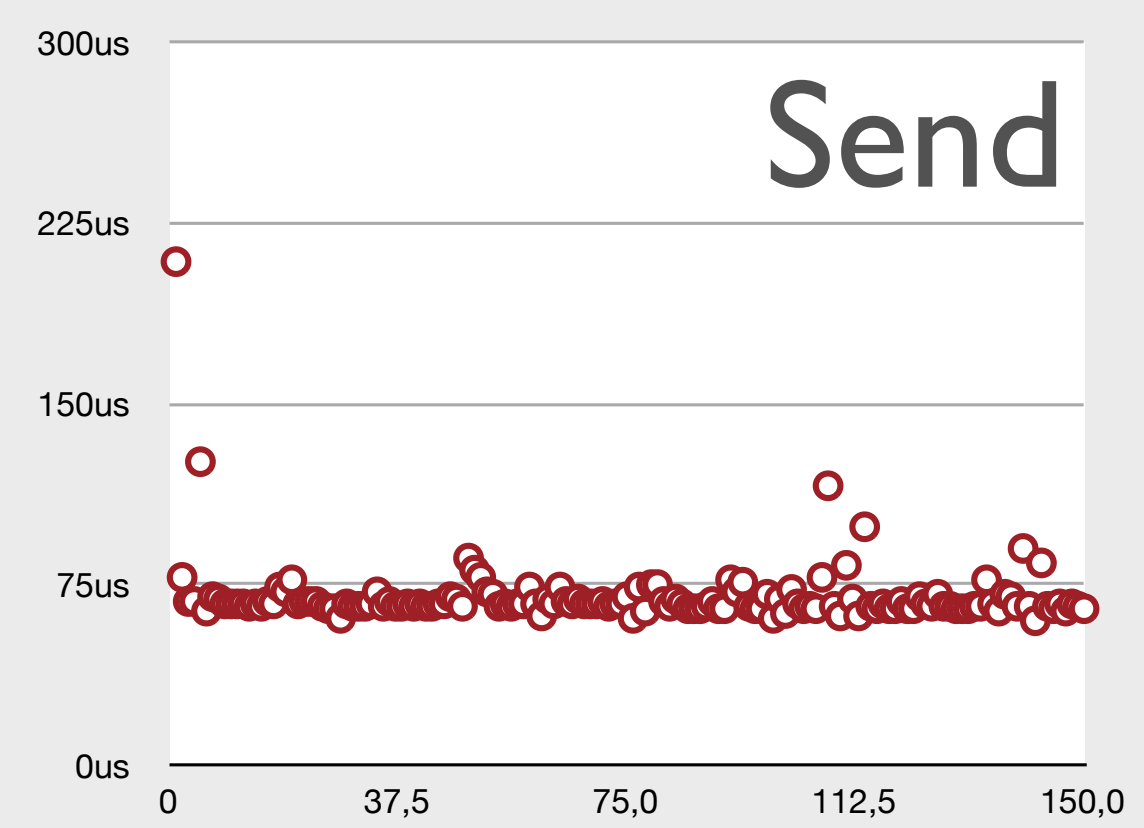
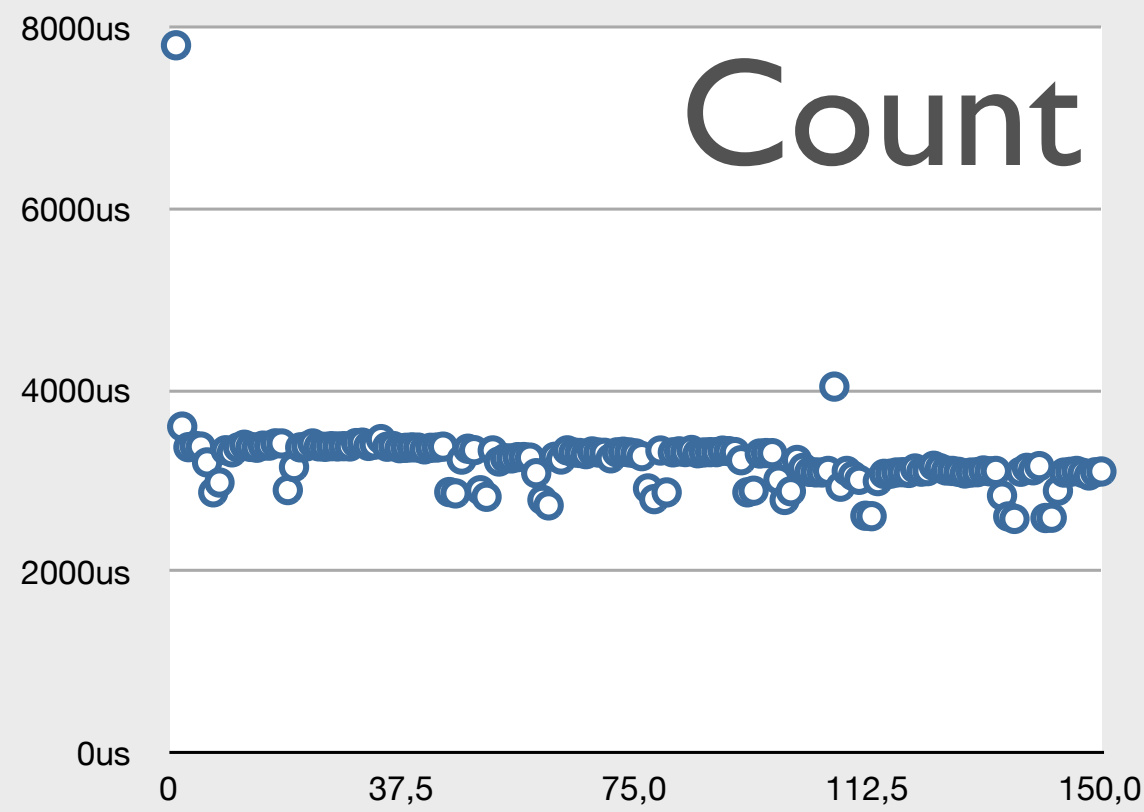


# COMPOUND

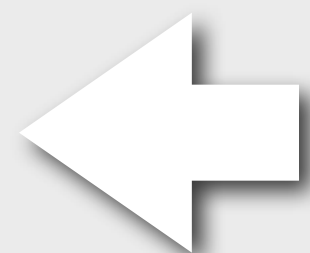
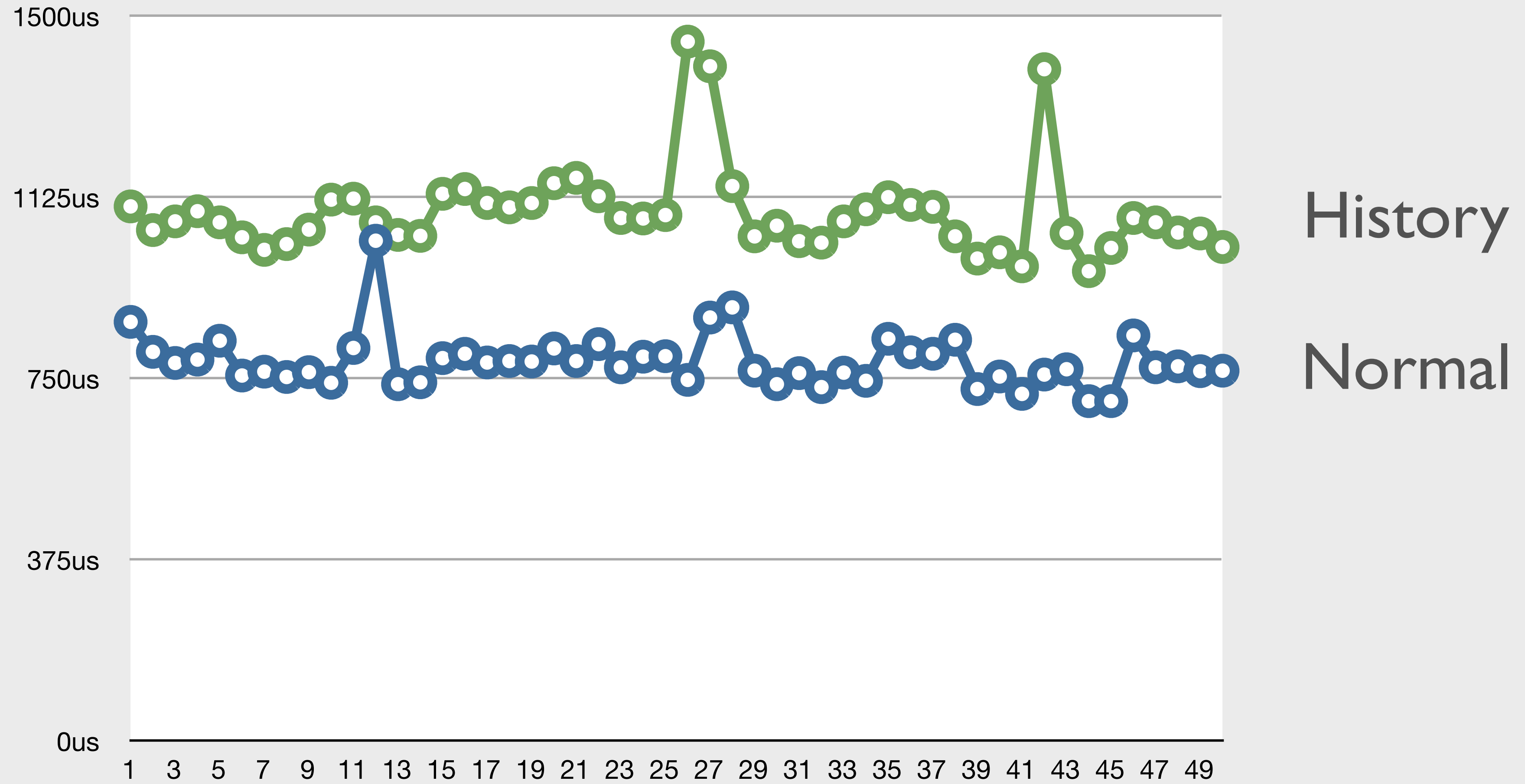
Only MIN,AVG



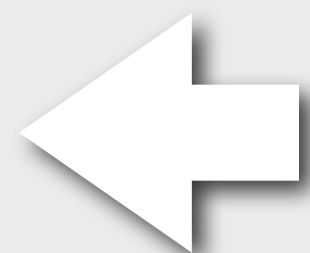
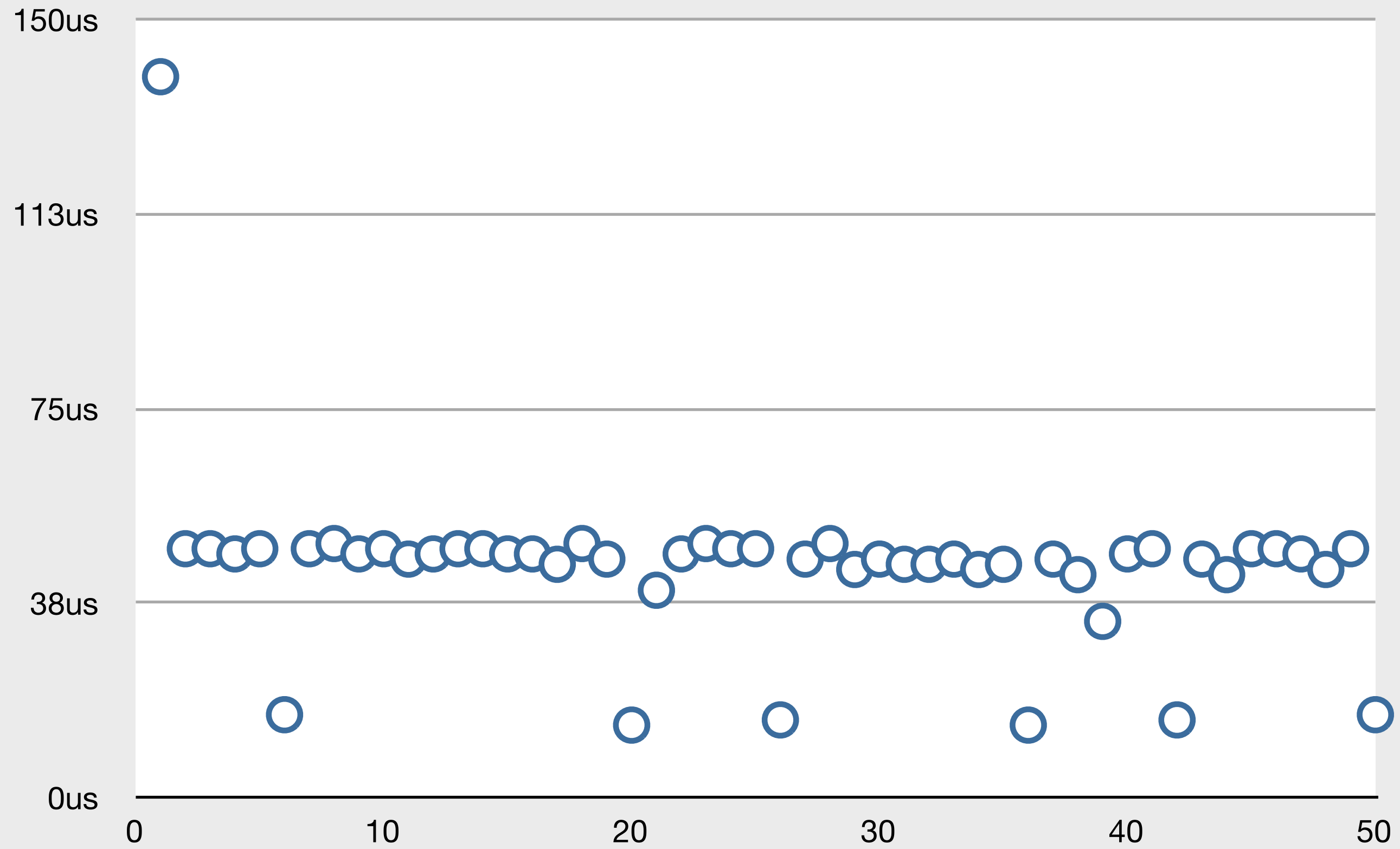
With MAX values



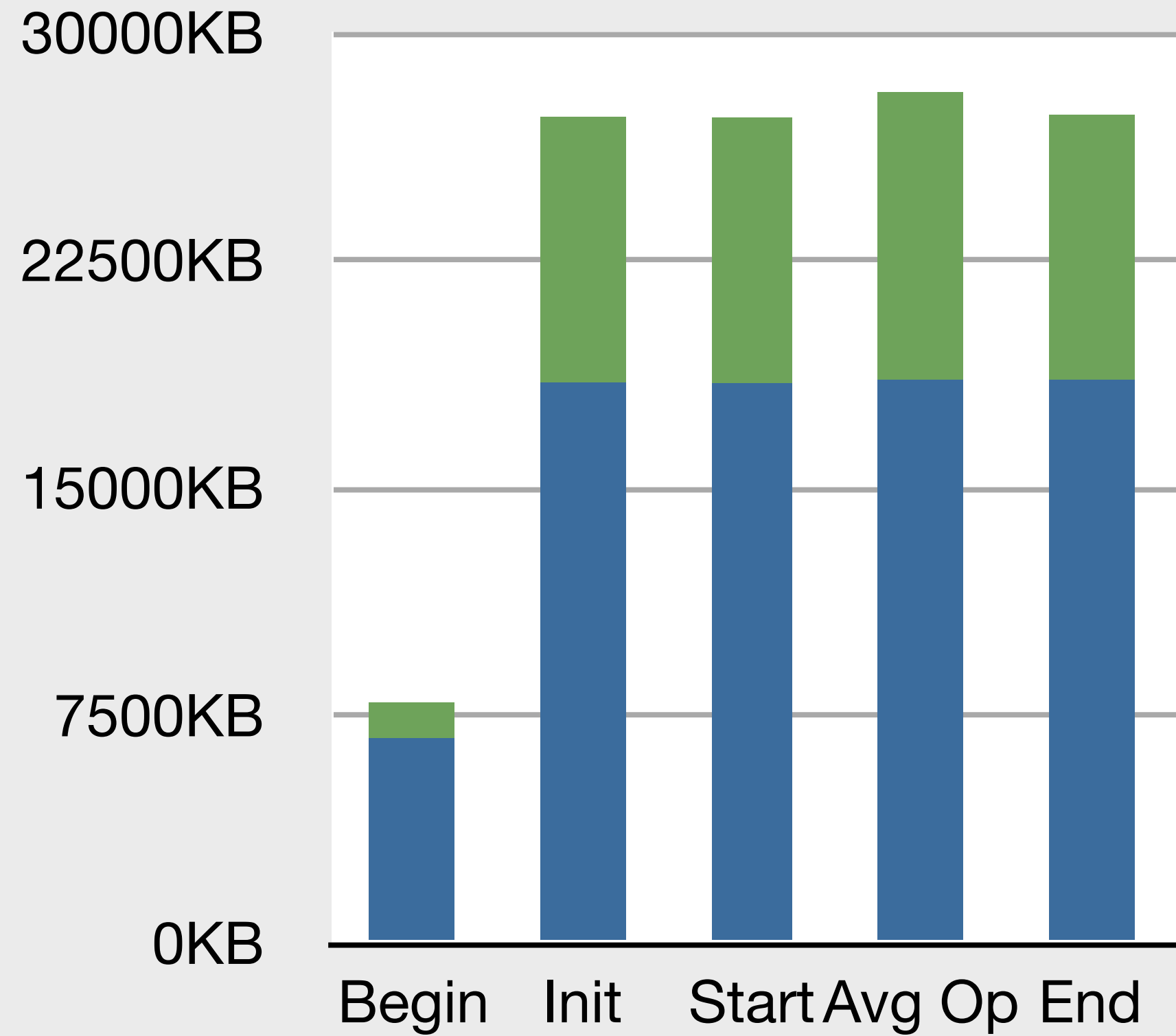
# CONCURRENT



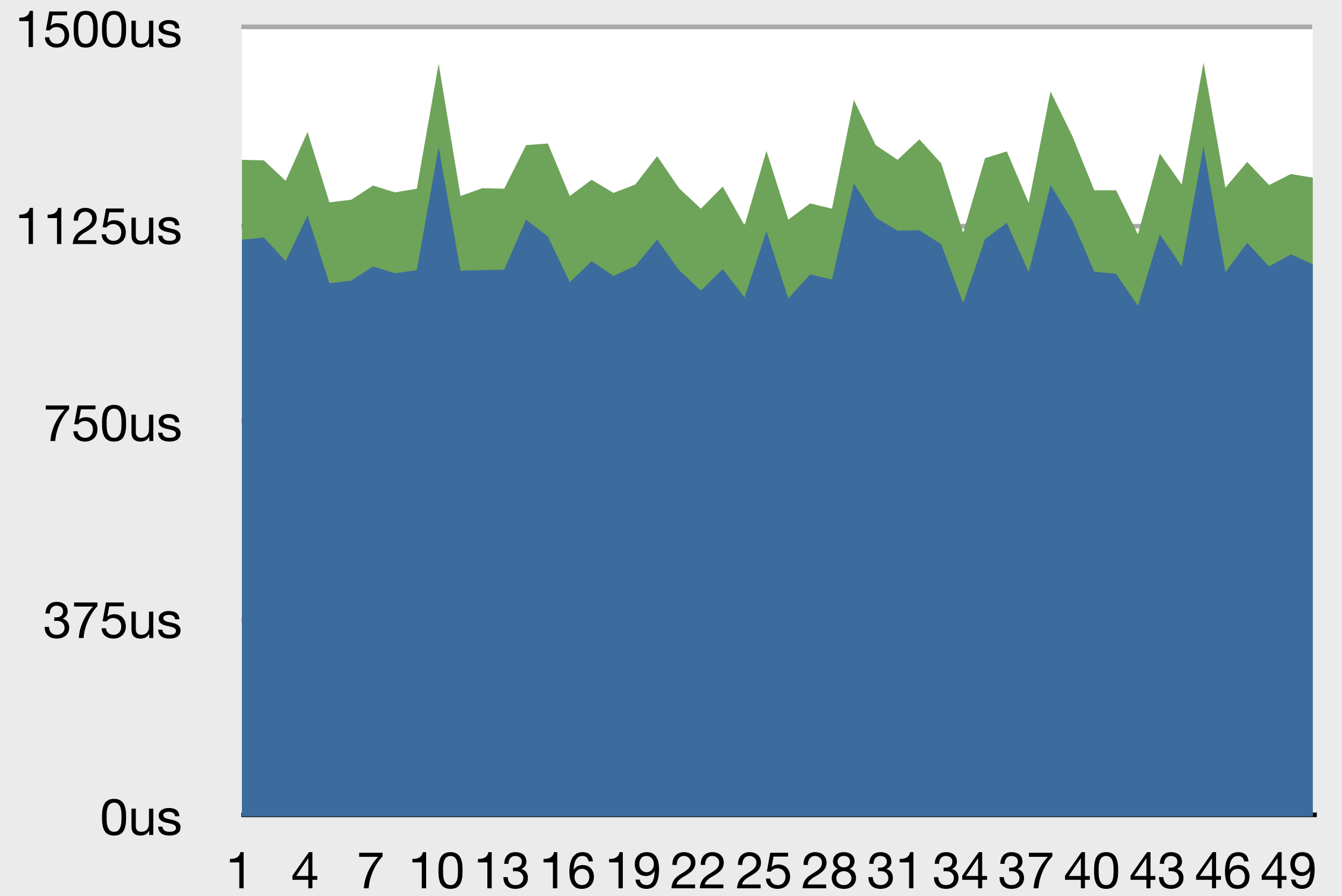
# CONFIG



# LIFECYCLE



Memory



Overall Delay

