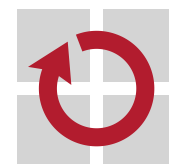
Systemgewahre statische Laufzeitanalyse von Universalbetriebssystemen

Simon Schuster

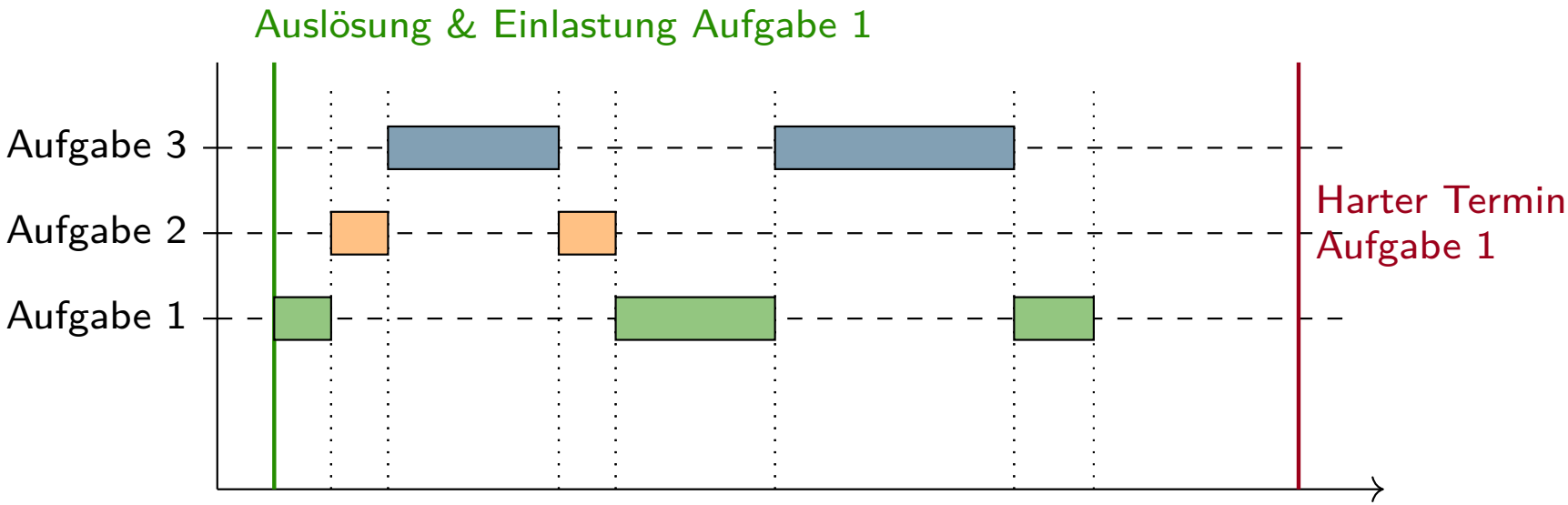
Betreuer:
Peter Ulbrich
Peter Wägemann
Wolfgang Schröder-Preikschat

18. Oktober 2018

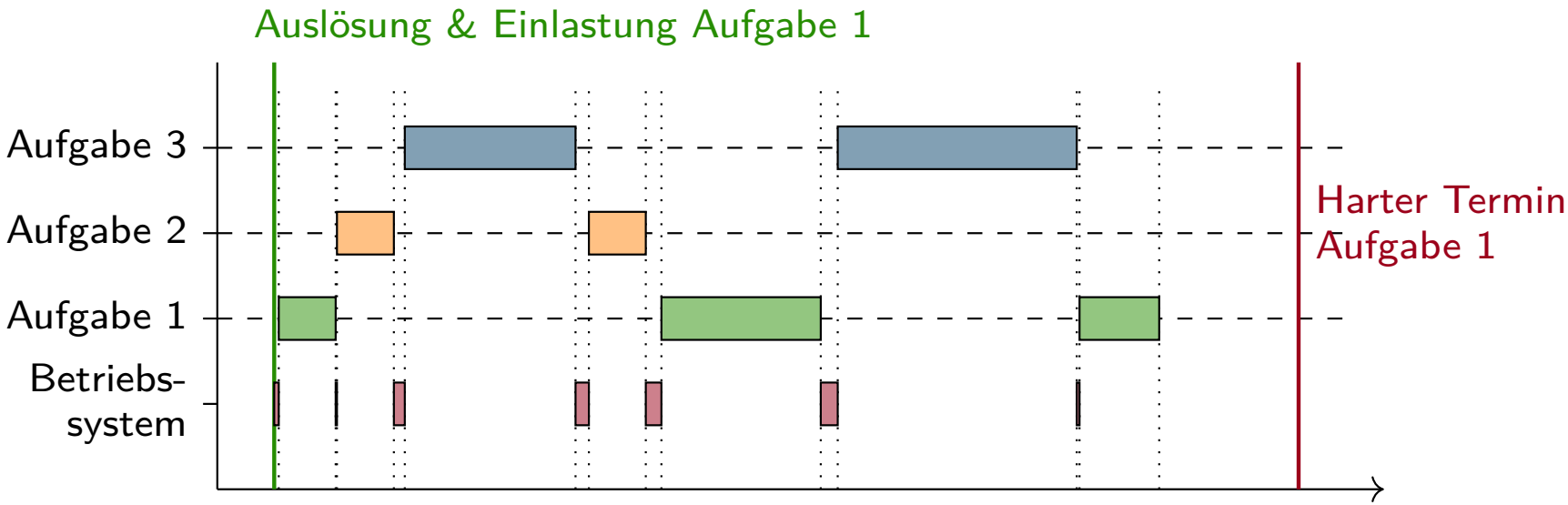
Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme
Friedrich-Alexander-Universität
Erlangen-Nürnberg



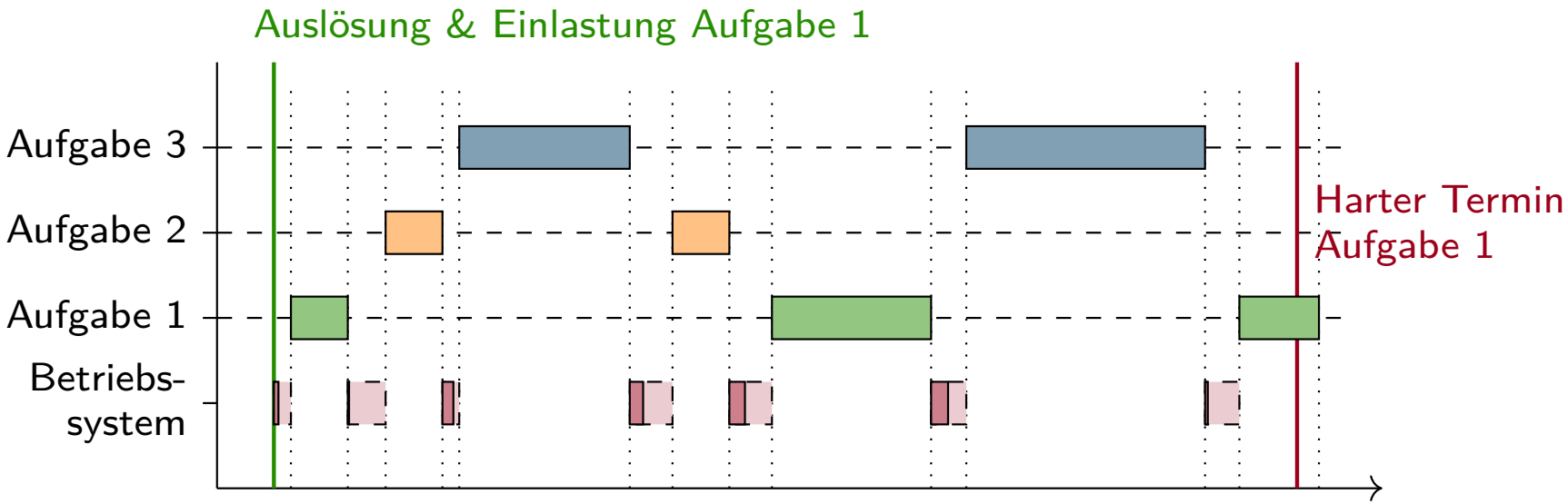
Betriebssystemvorgänge und Termintreue



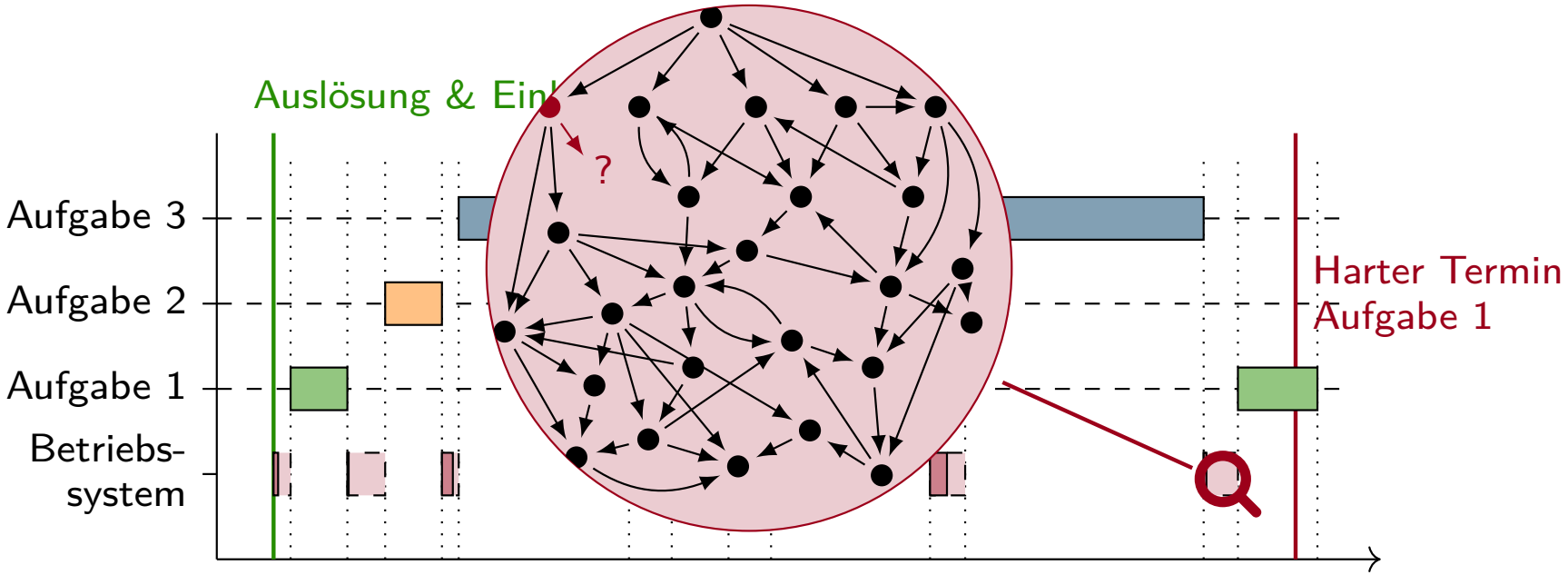
Betriebssystemvorgänge und Termintreue



Betriebssystemvorgänge und Termintreue

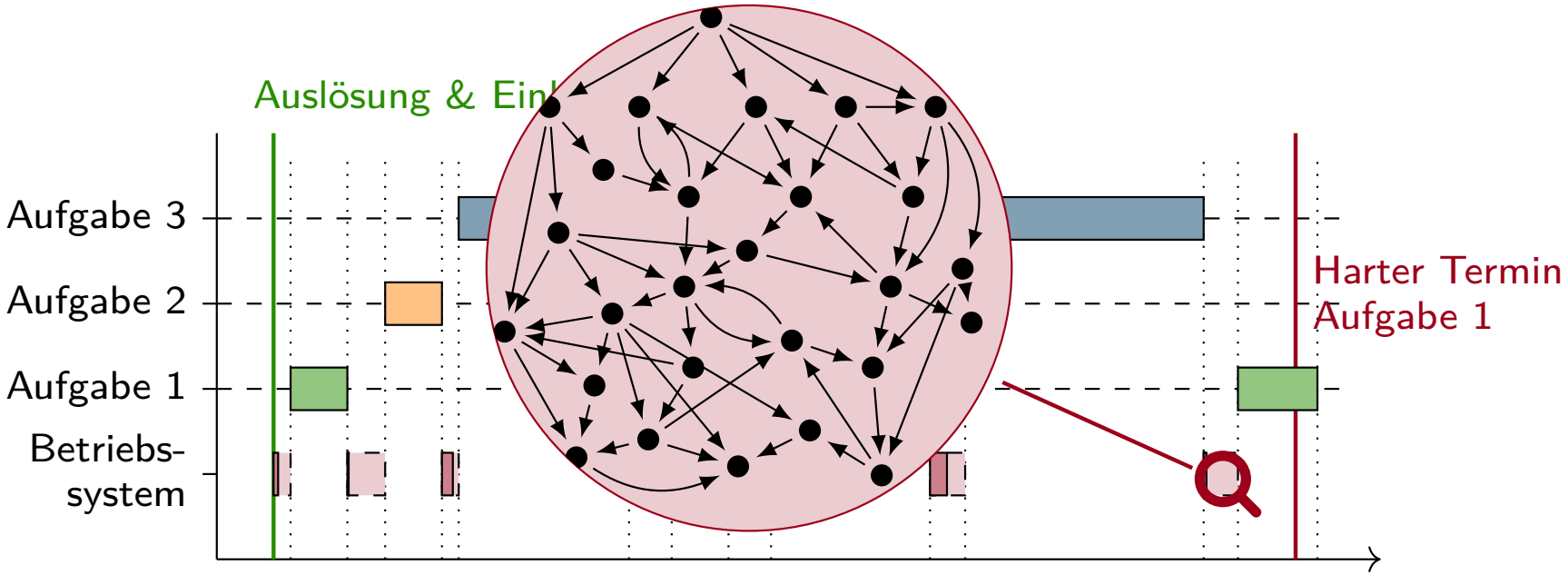


Betriebssystemvorgänge und Termintreue



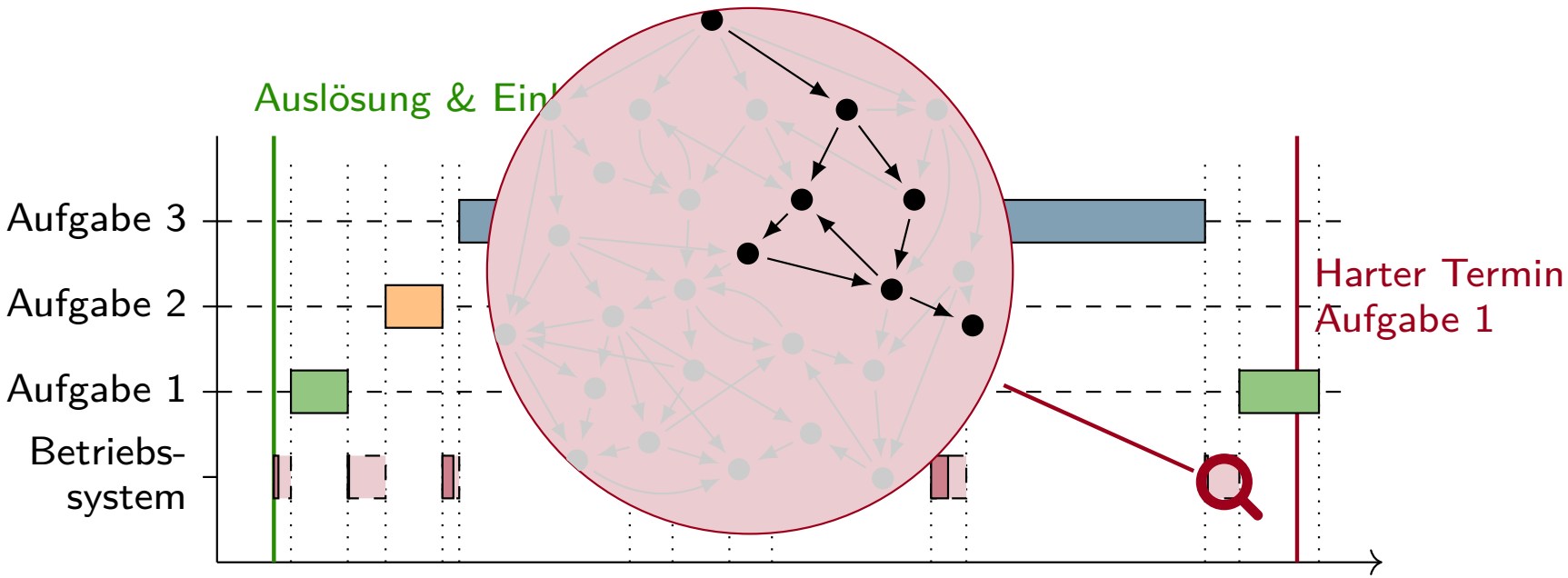
✗ **Fehlschlagende Kontrollflussrekonstruktion** durch Funktionszeiger

Betriebssystemvorgänge und Termintreue



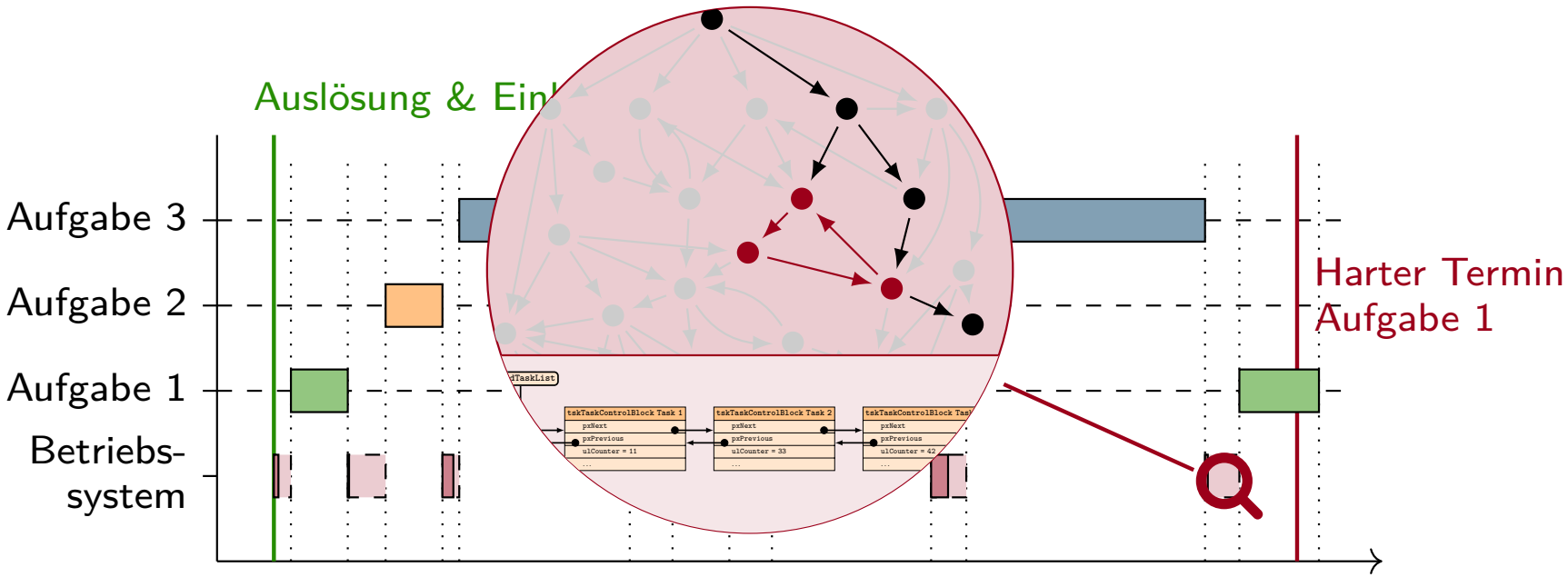
✗ **Fehlschlagende Kontrollflussrekonstruktion** durch Funktionszeiger

Betriebssystemvorgänge und Termintreue



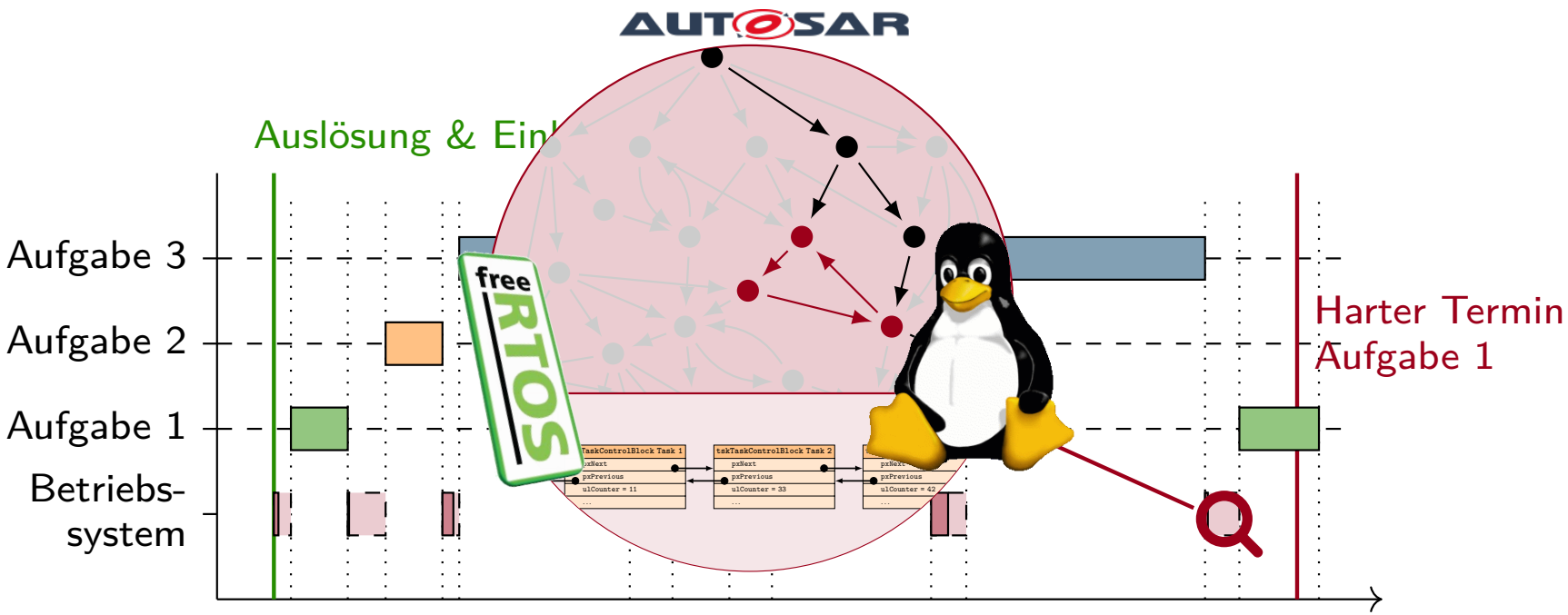
- ✗ **Fehlschlagende Kontrollflussrekonstruktion** durch Funktionszeiger
- ✗ **Hoher Pessimismus** mangels kontextsensitiver Pfadidentifikation

Betriebssystemvorgänge und Termintreue



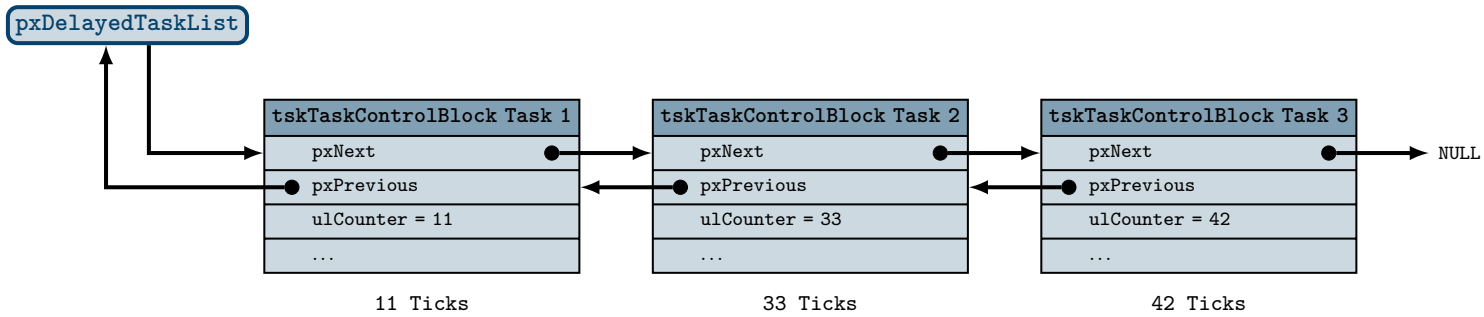
- ✗ **Fehlschlagende Kontrollflussrekonstruktion** durch Funktionszeiger
- ✗ **Hoher Pessimismus** mangels kontextsensitiver Pfadidentifikation
- ✗ **Unbeschränkte WCET** aufgrund anwendungsabhängiger Datenstrukturen

Betriebssystemvorgänge und Termintreue

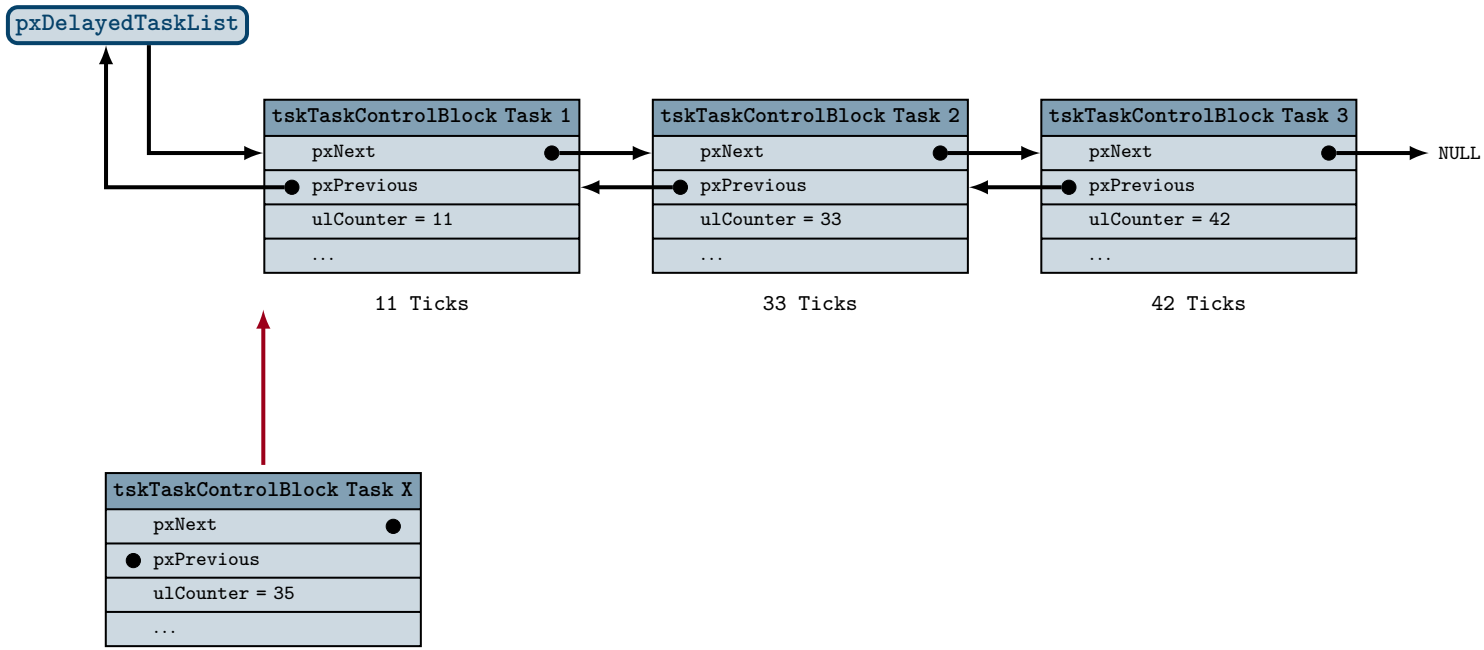


- ✗ **Fehlschlagende Kontrollflussrekonstruktion** durch Funktionszeiger
- ✗ **Hoher Pessimismus** mangels kontextsensitiver Pfadidentifikation
- ✗ **Unbeschränkte WCET** aufgrund anwendungsabhängiger Datenstrukturen

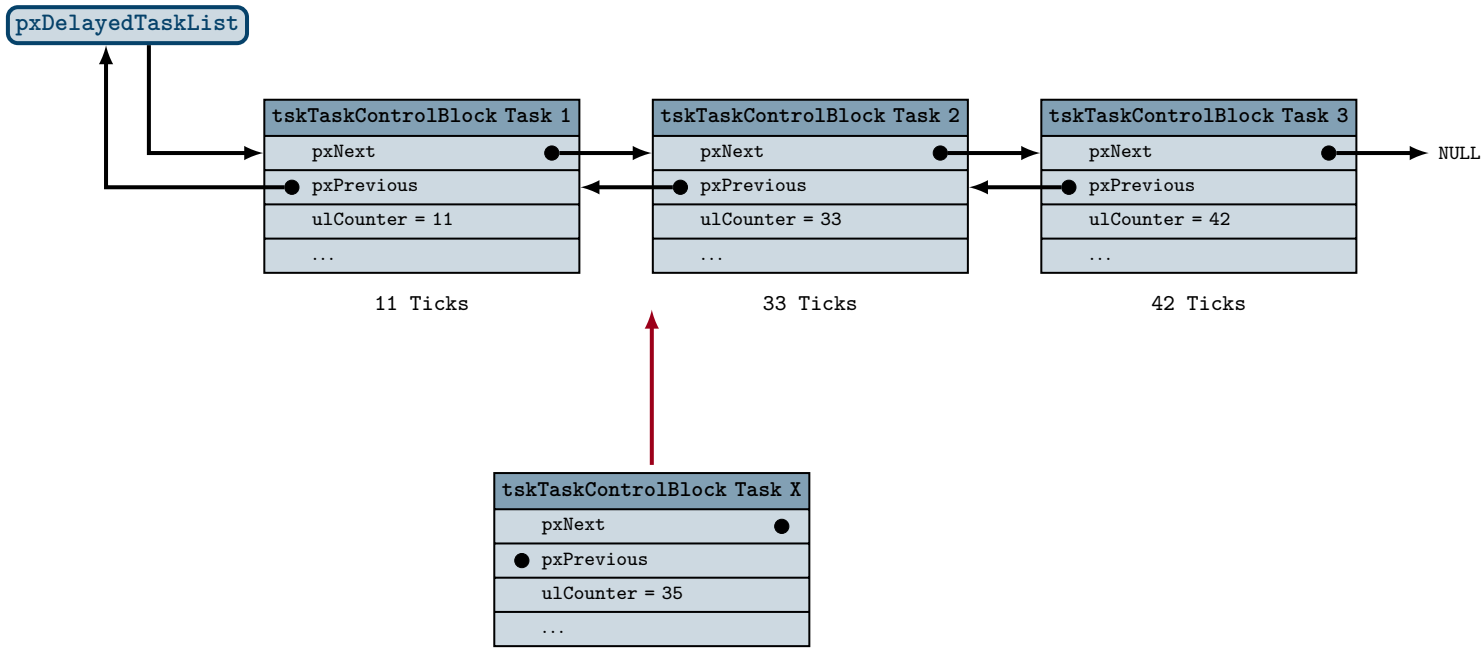
Anatomie von Betriebssystemaufrufen



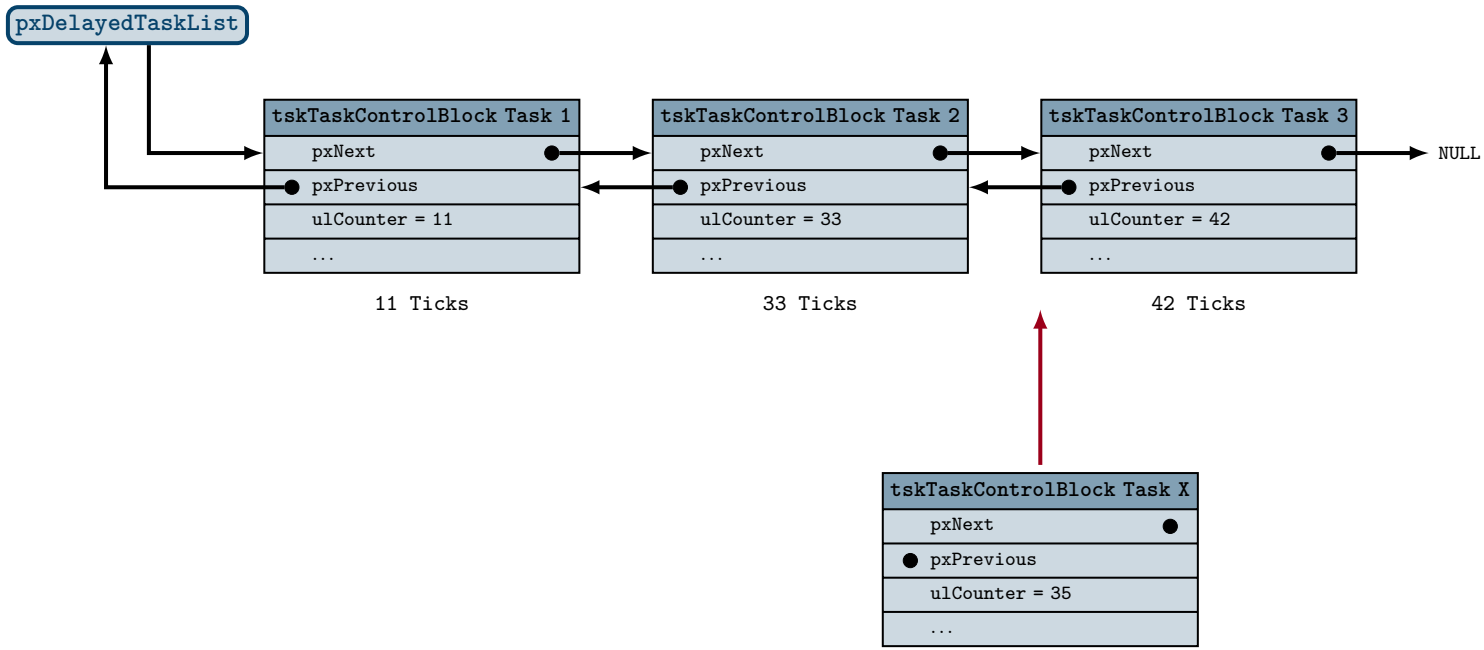
Anatomie von Betriebssystemaufrufen



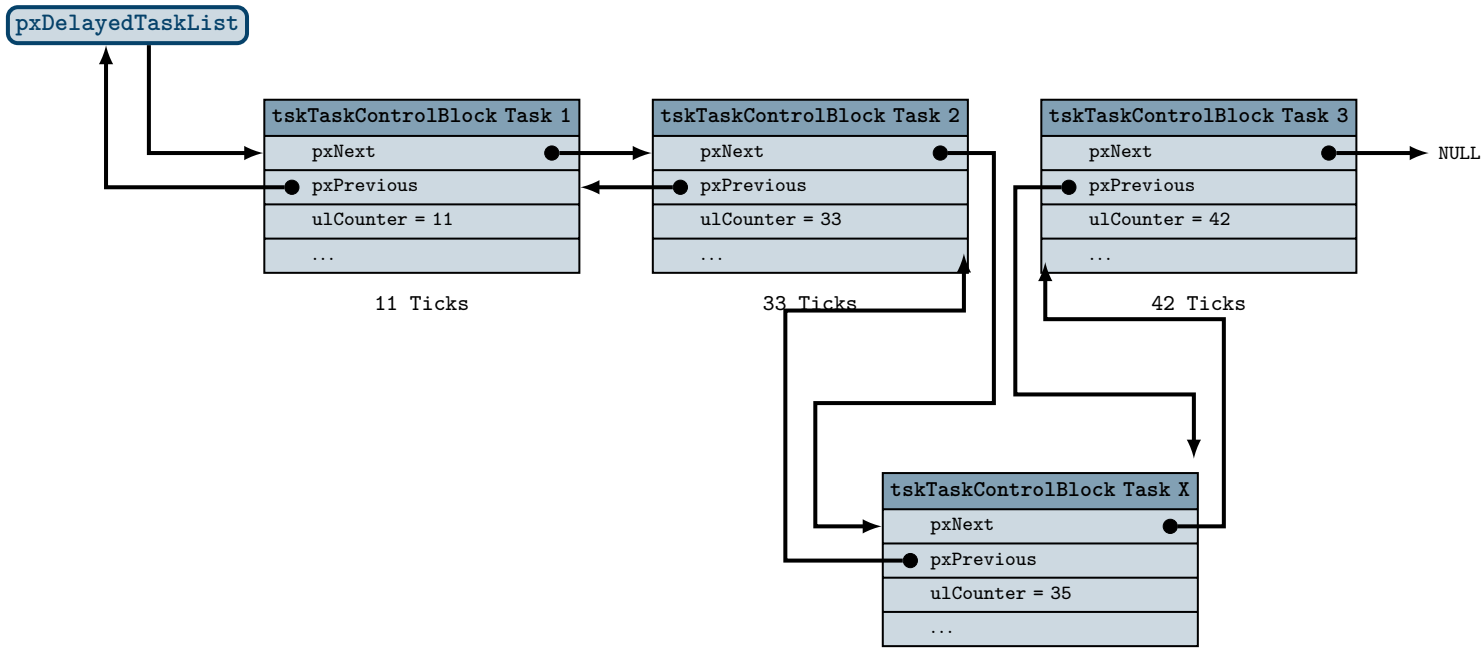
Anatomie von Betriebssystemaufrufen



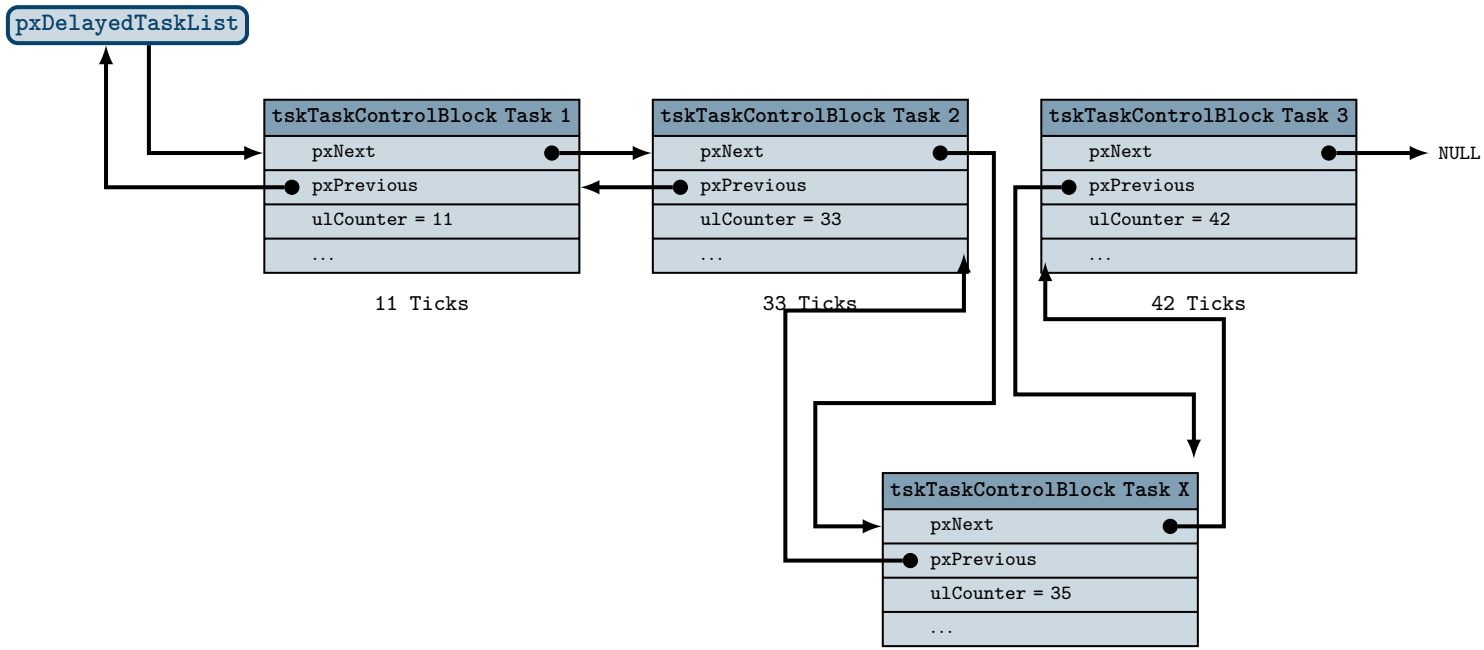
Anatomie von Betriebssystemaufrufen



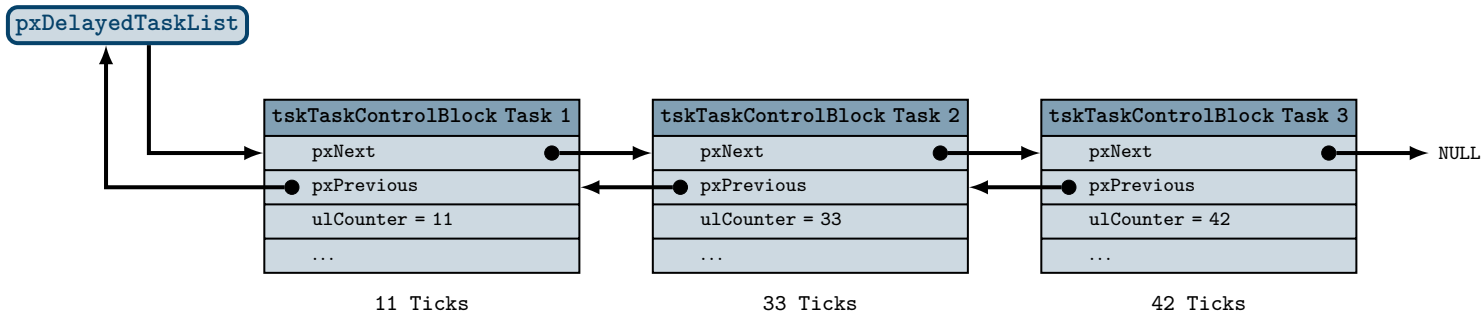
Anatomie von Betriebssystemaufrufen

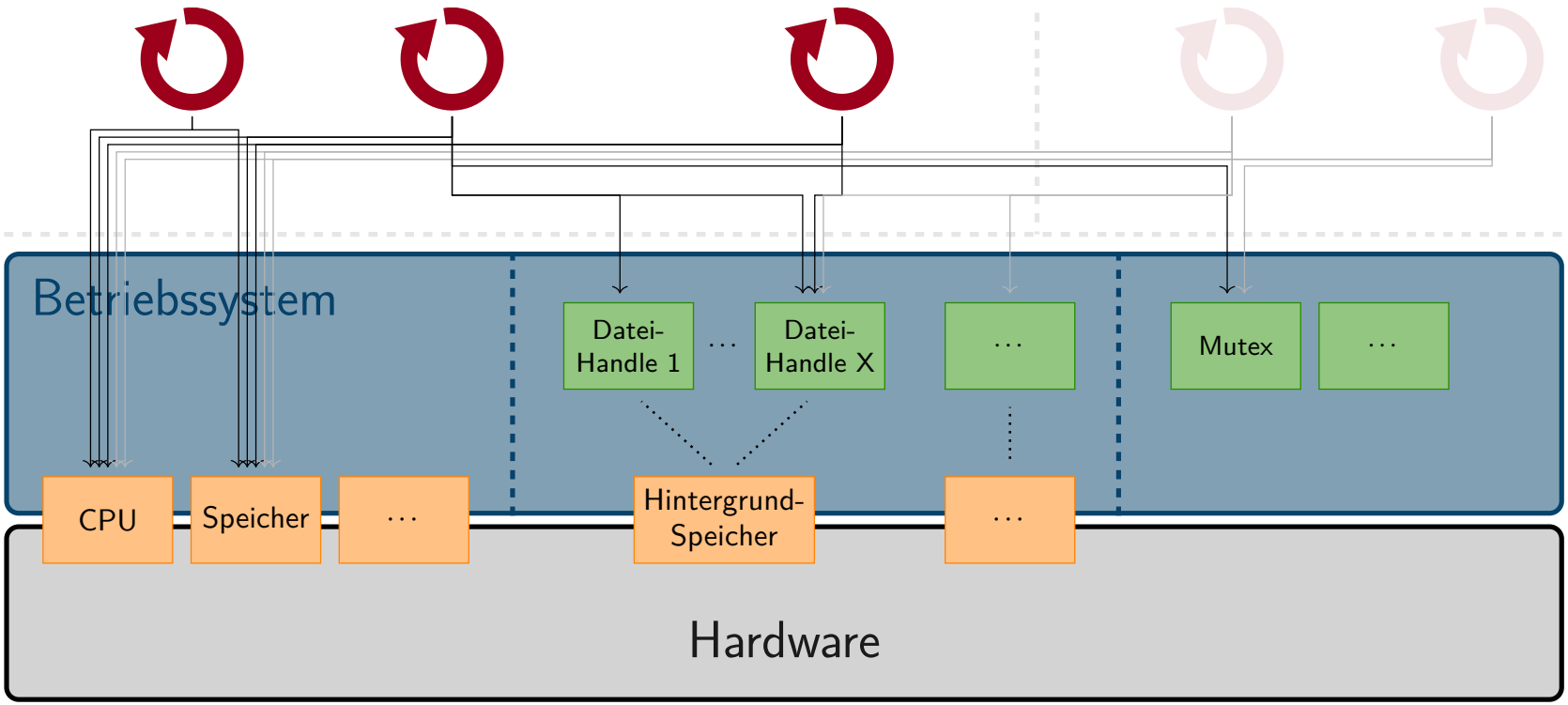


Anatomie von Betriebssystemaufrufen

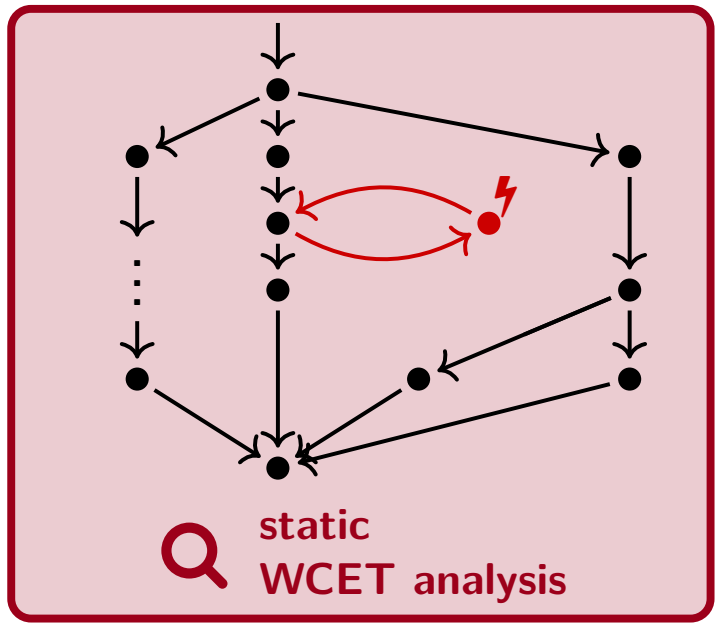


Anatomie von Betriebssystemaufrufen

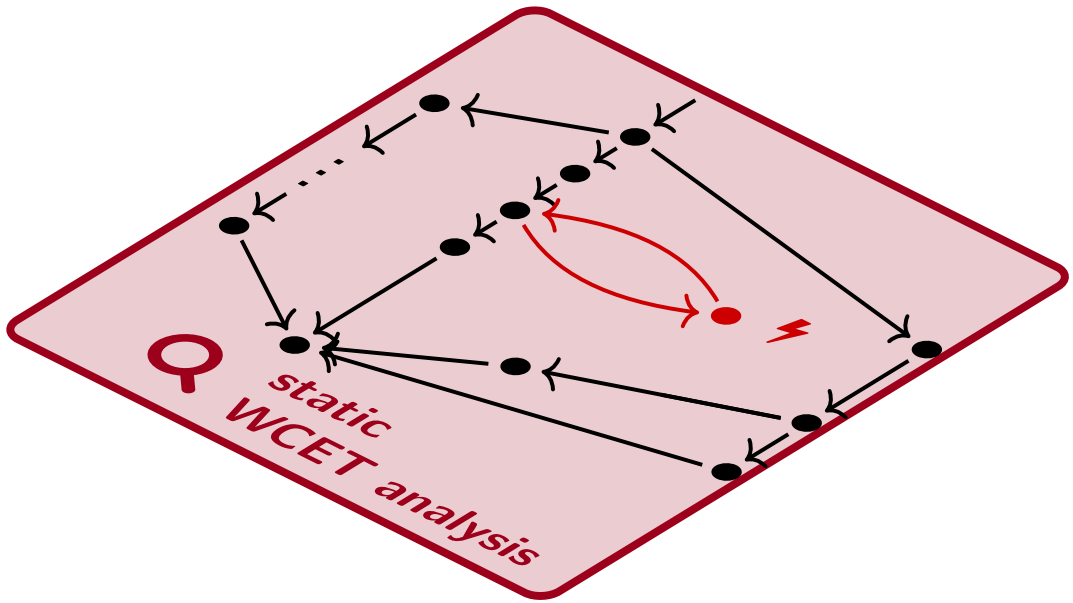




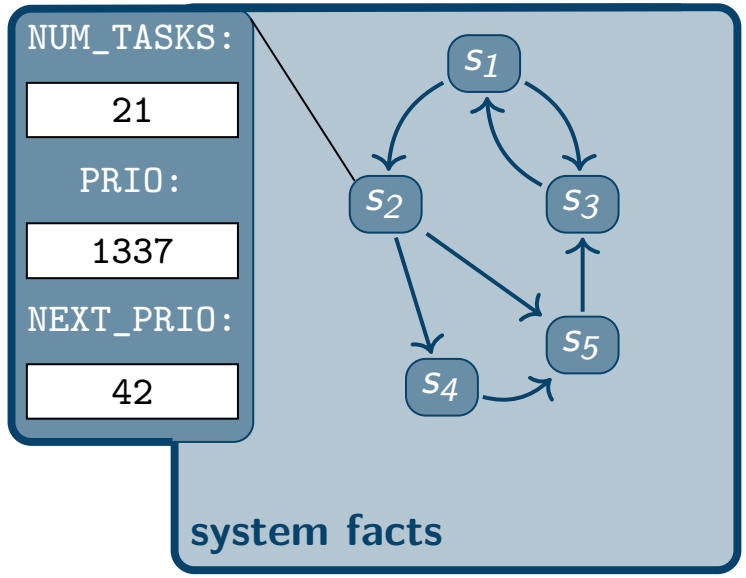
Statische Laufzeitanlyse von Betriebssystemaufrufen

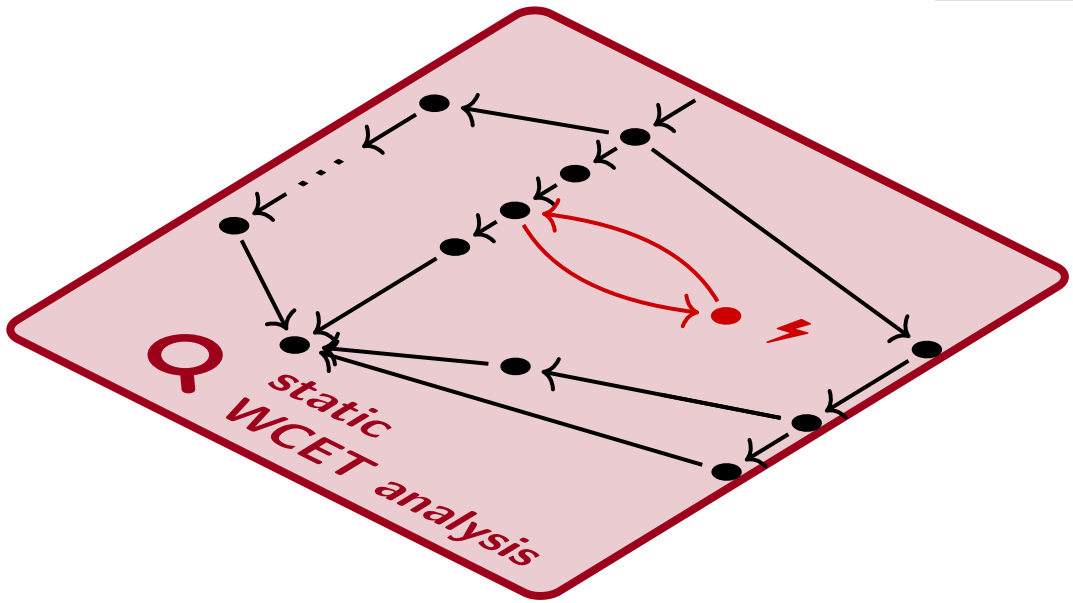


Statische Laufzeitanlyse von Betriebssystemaufrufen

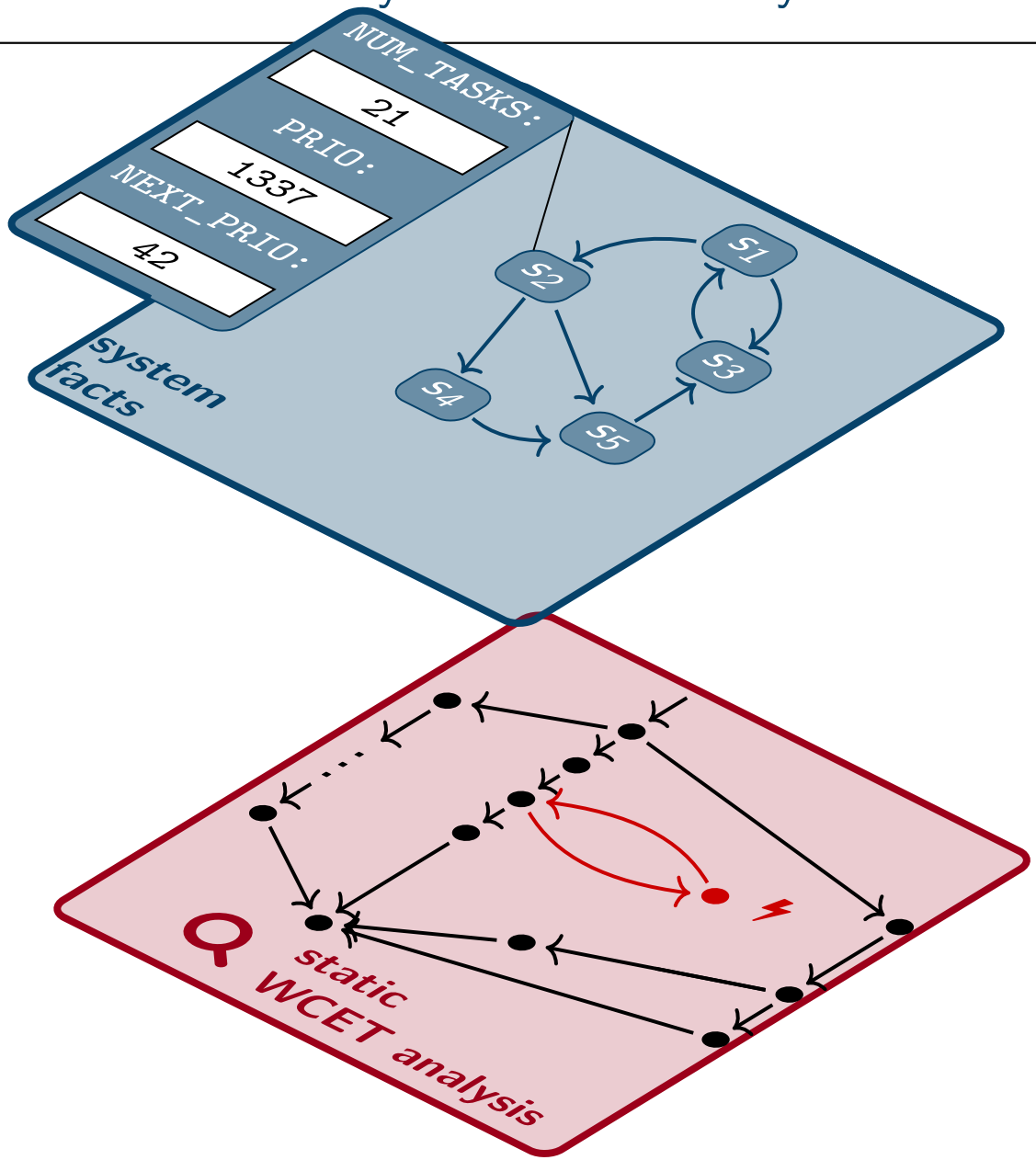


Statische Laufzeitanlyse von Betriebssystemaufrufen

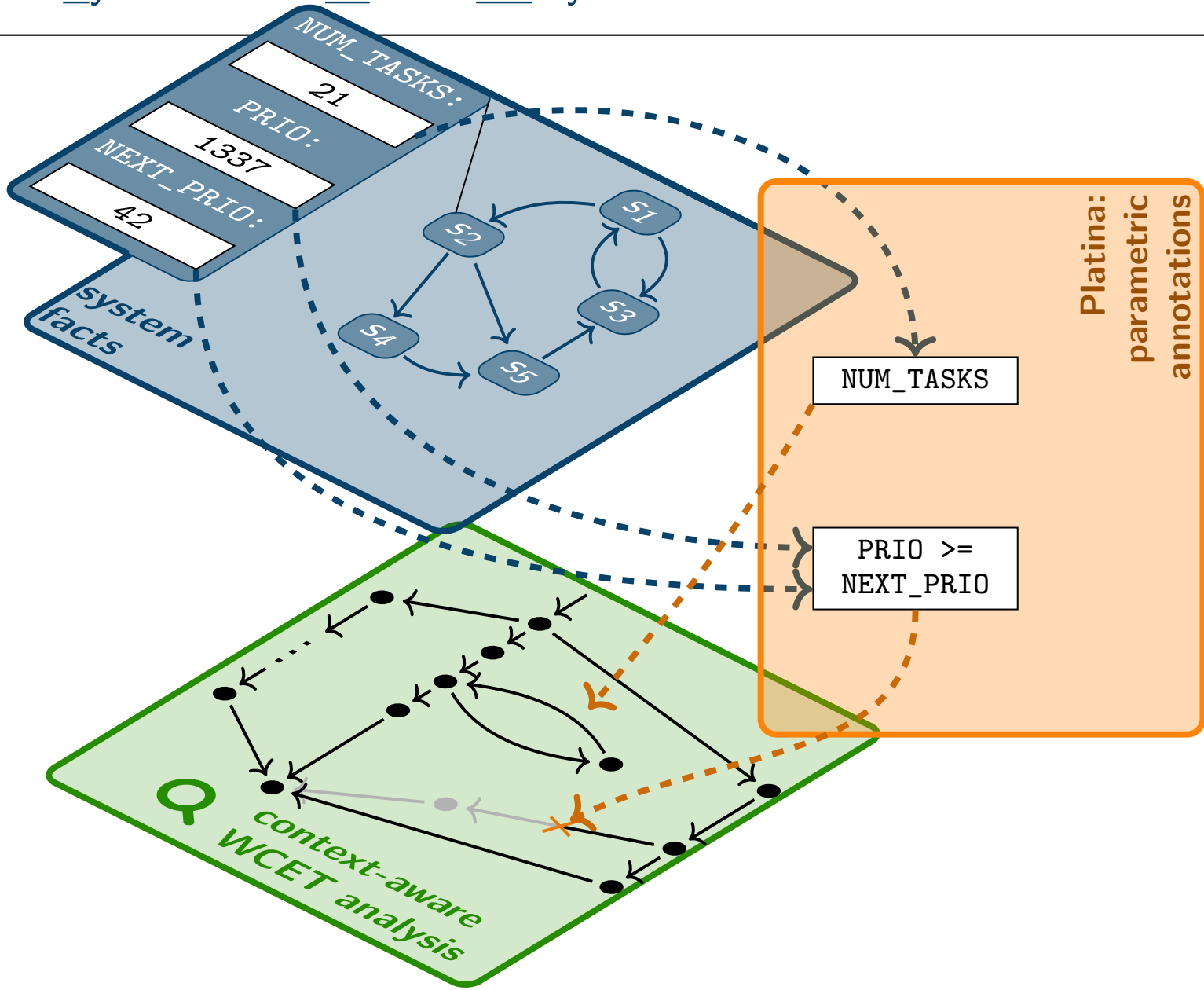




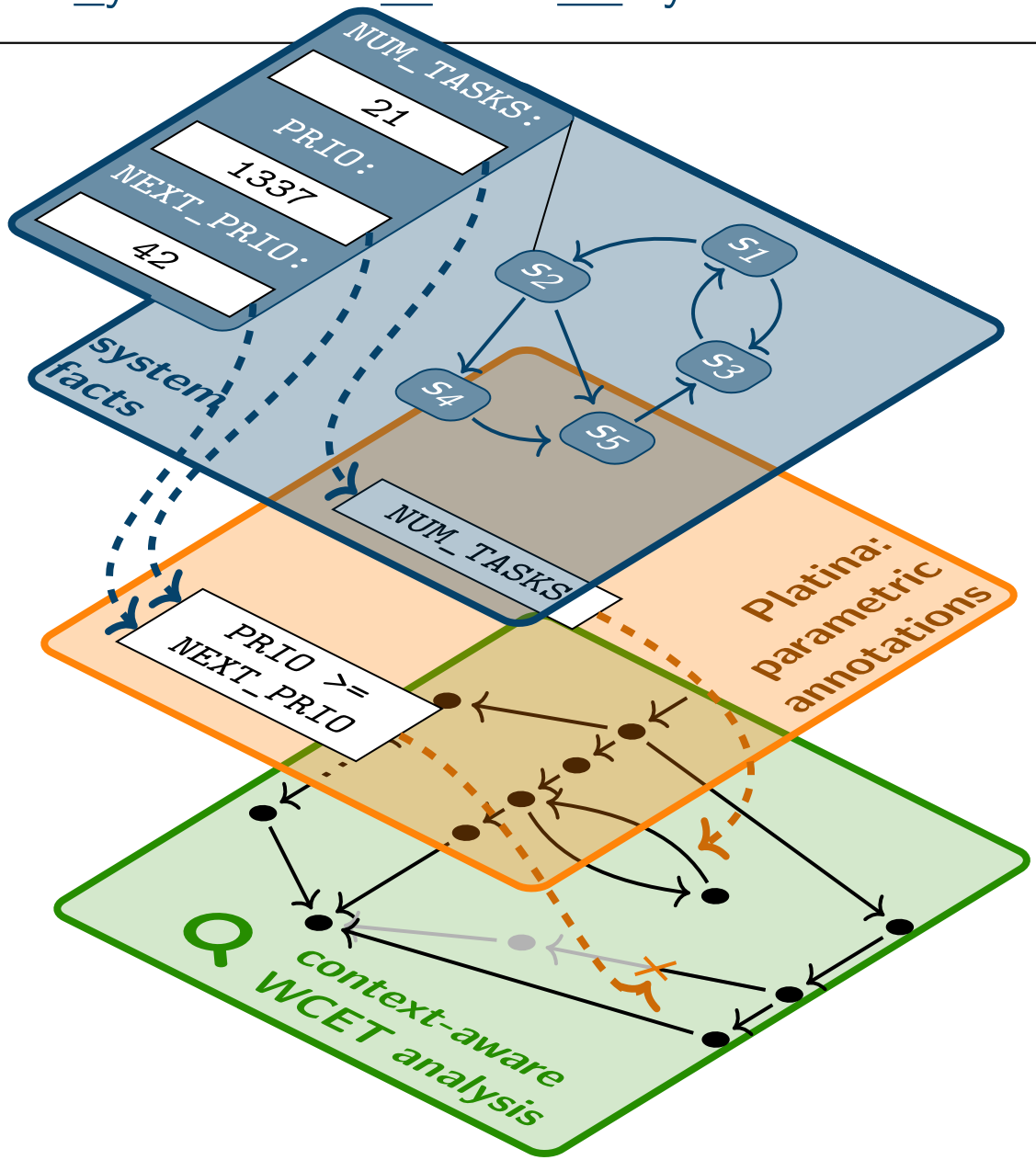
Statische Laufzeitanlyse von Betriebssystemaufrufen



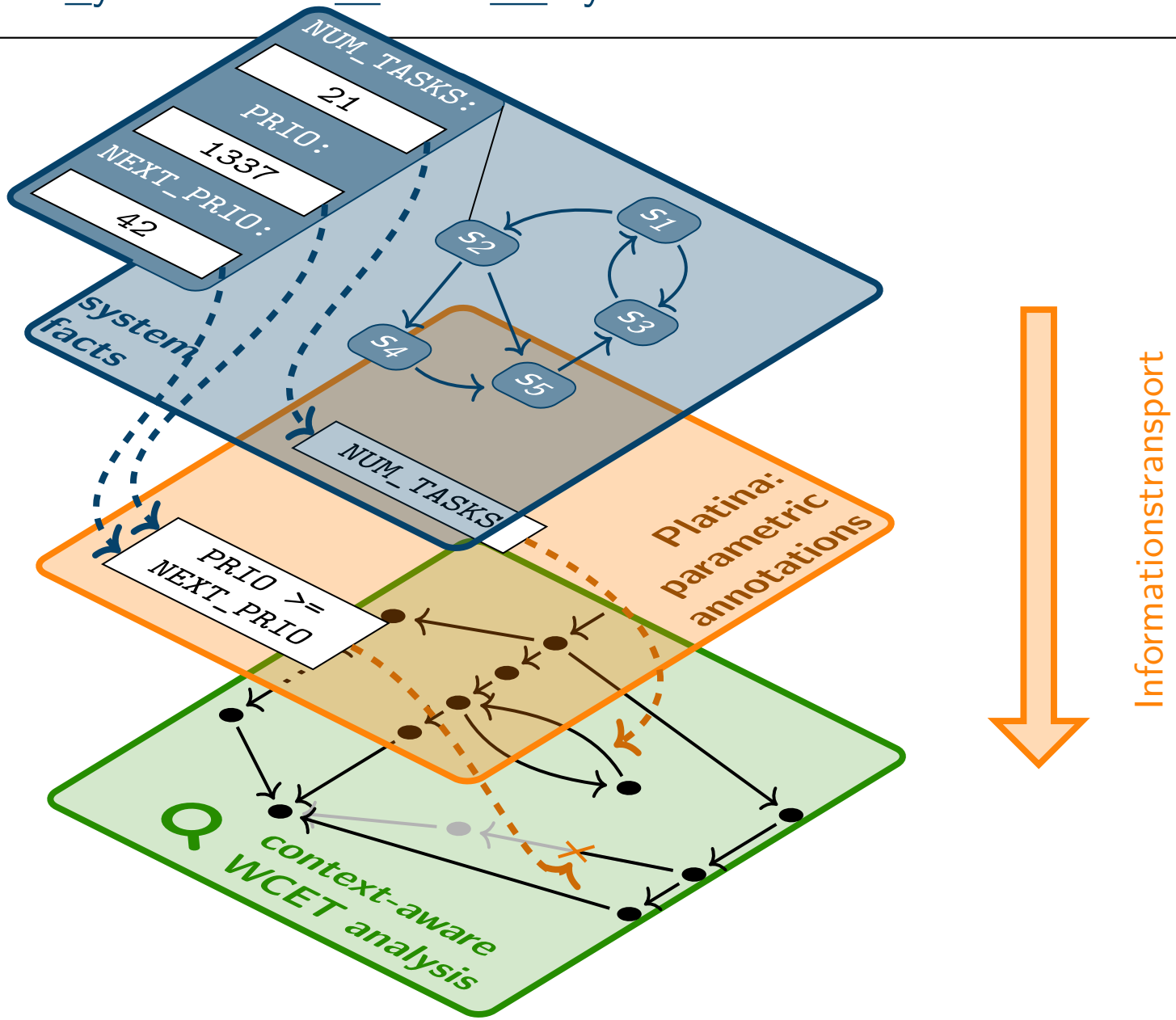
SWAN: System-wide WCET Analyzer



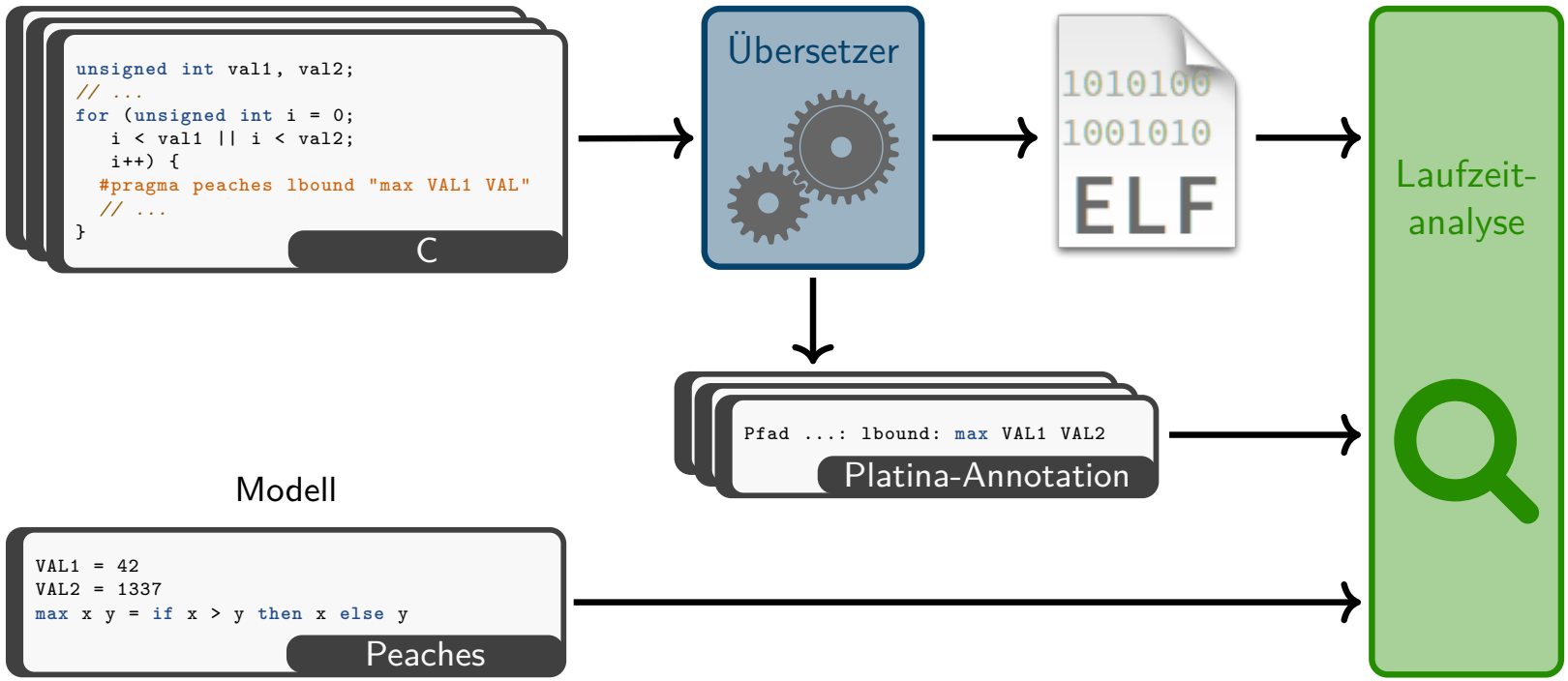
SWAN: System-wide WCET Analyzer



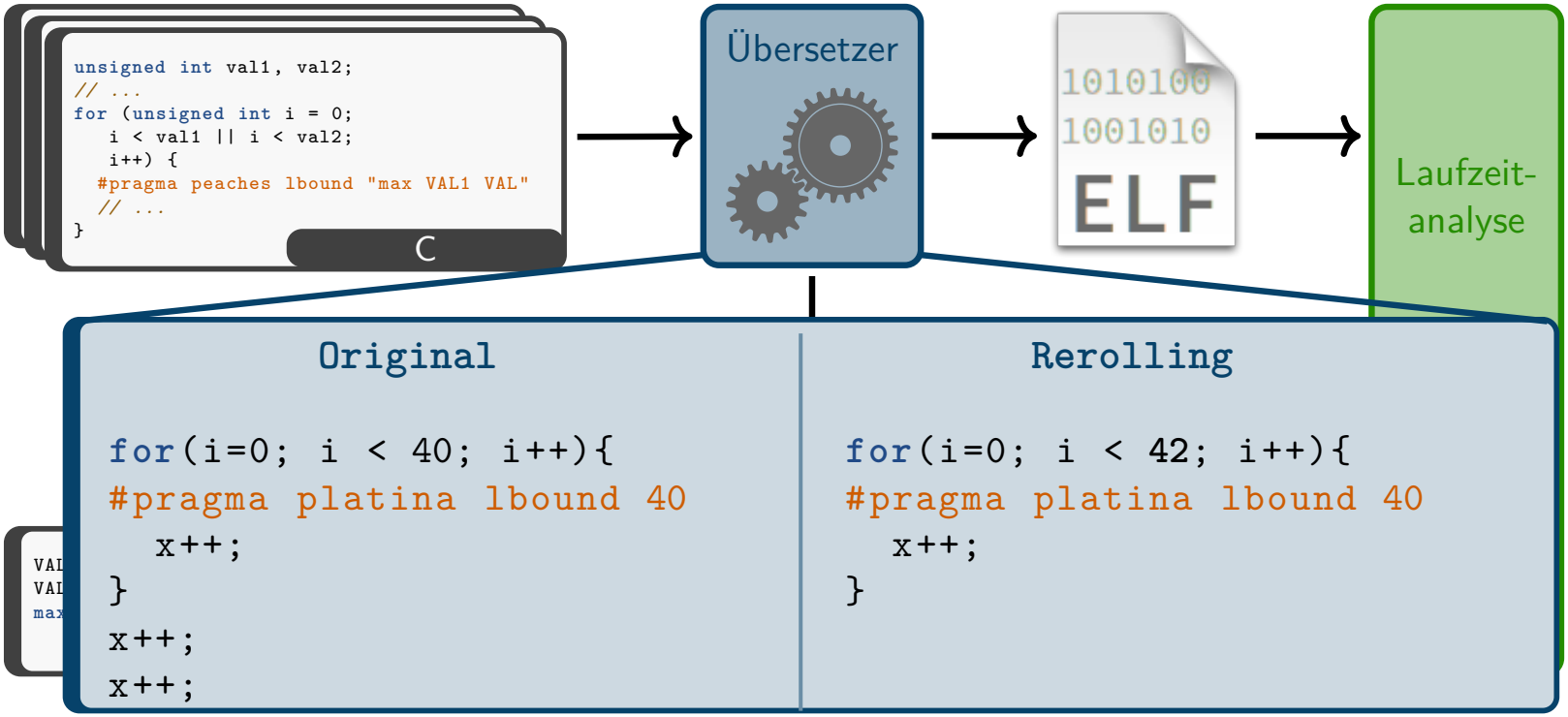
SWAN: System-wide WCET Analyzer



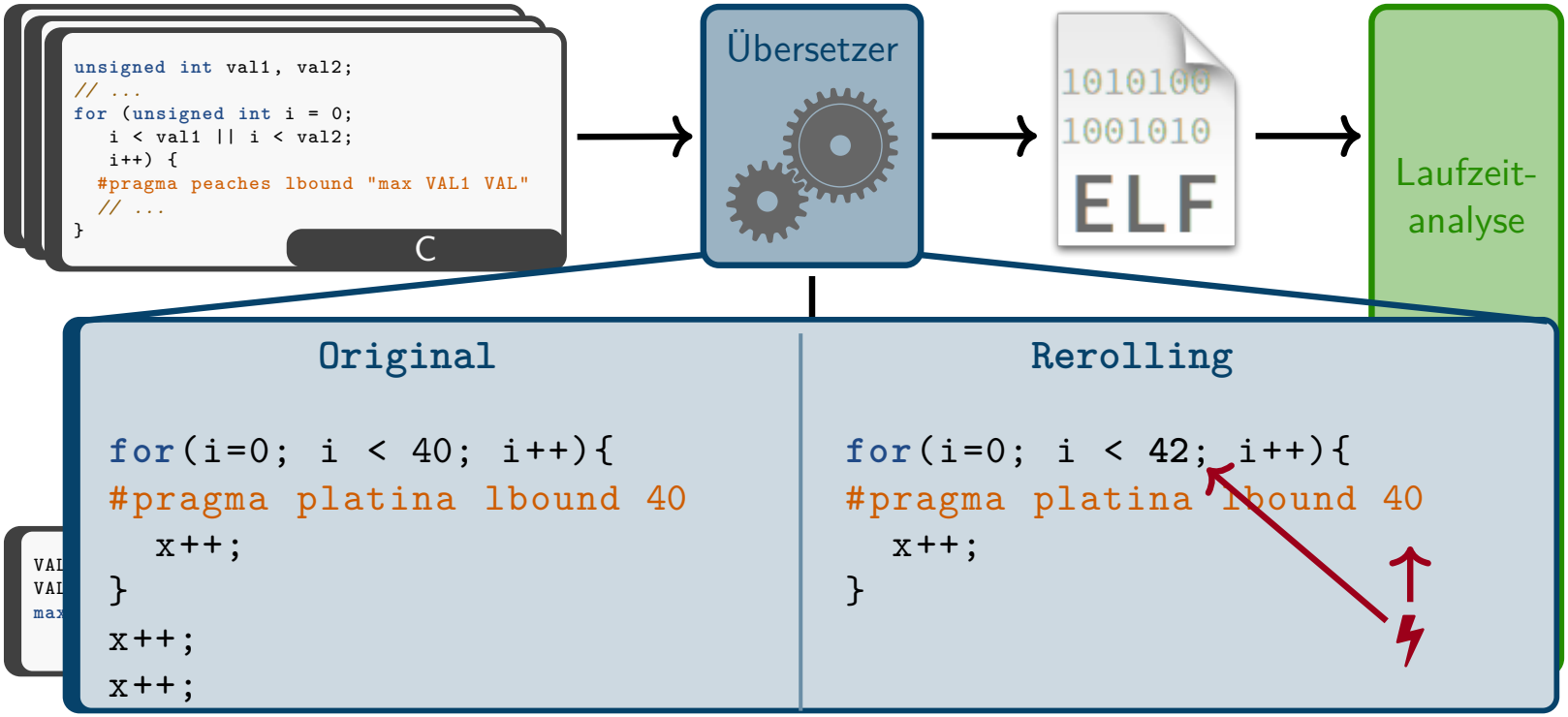
Architektur



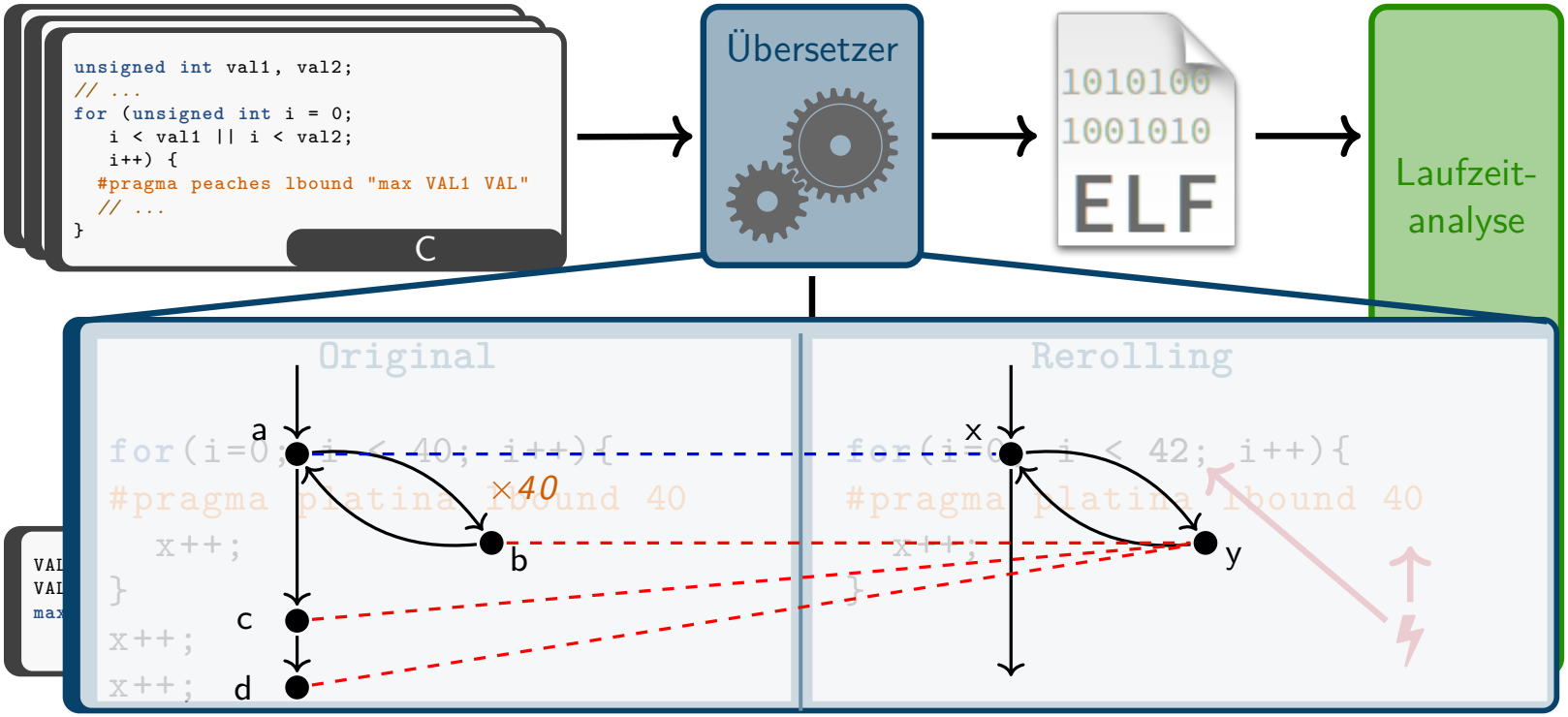
Architektur



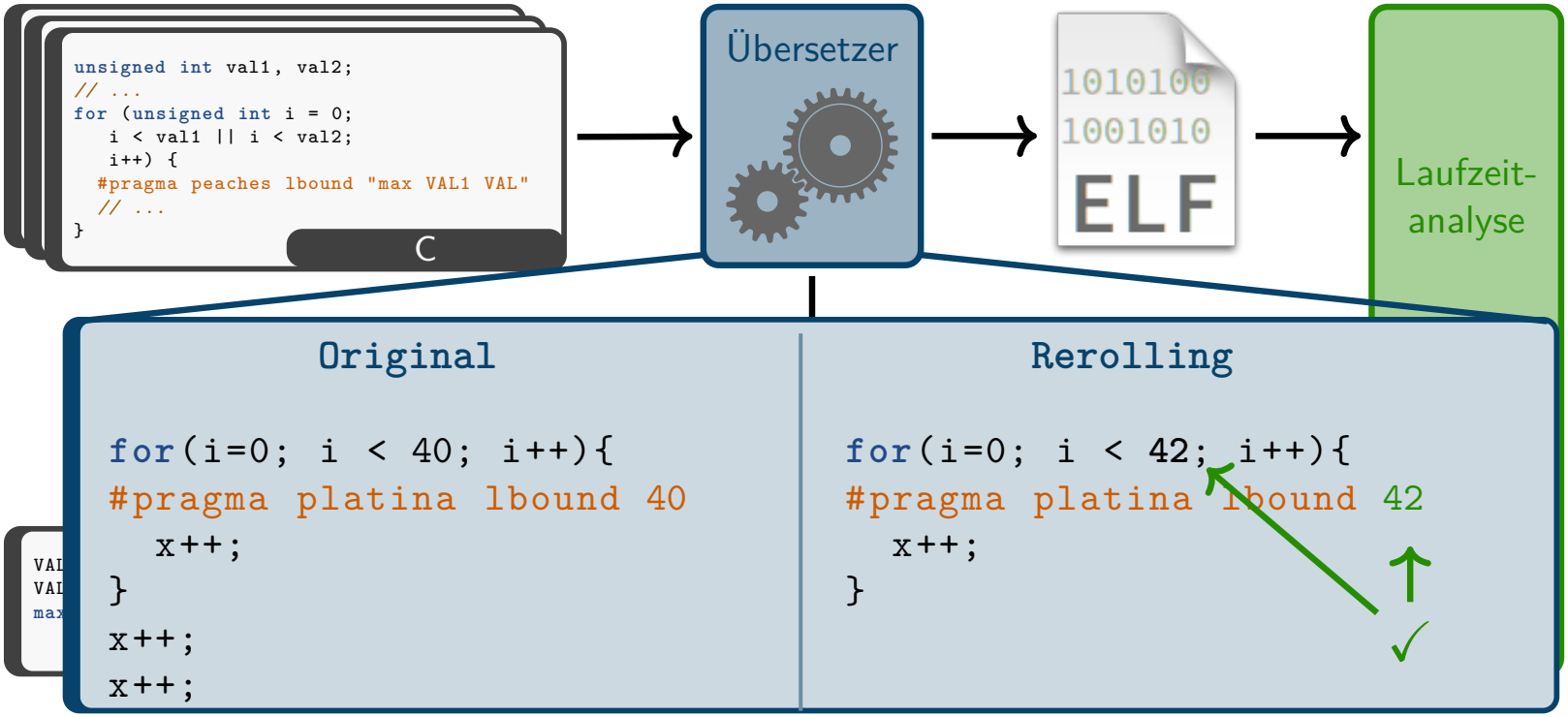
Architektur



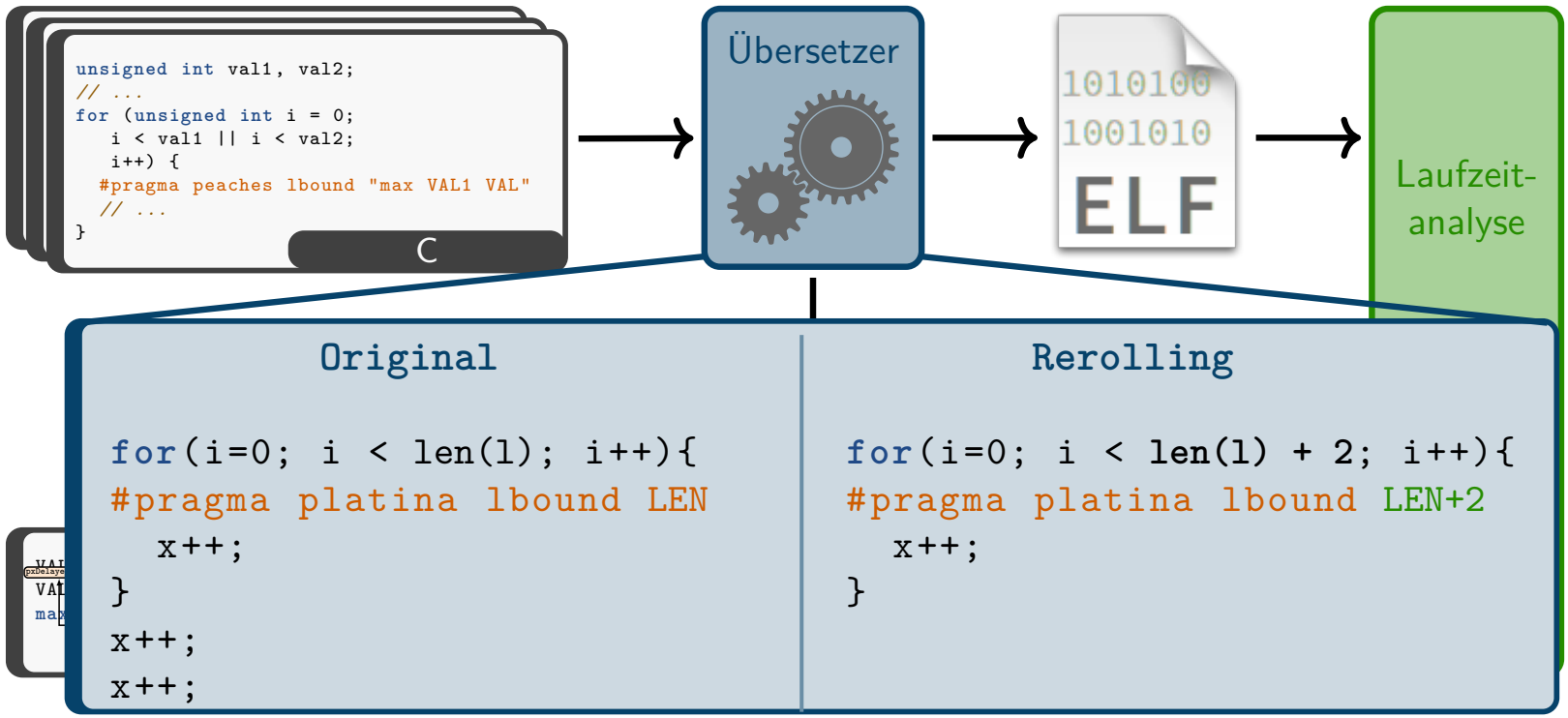
Architektur



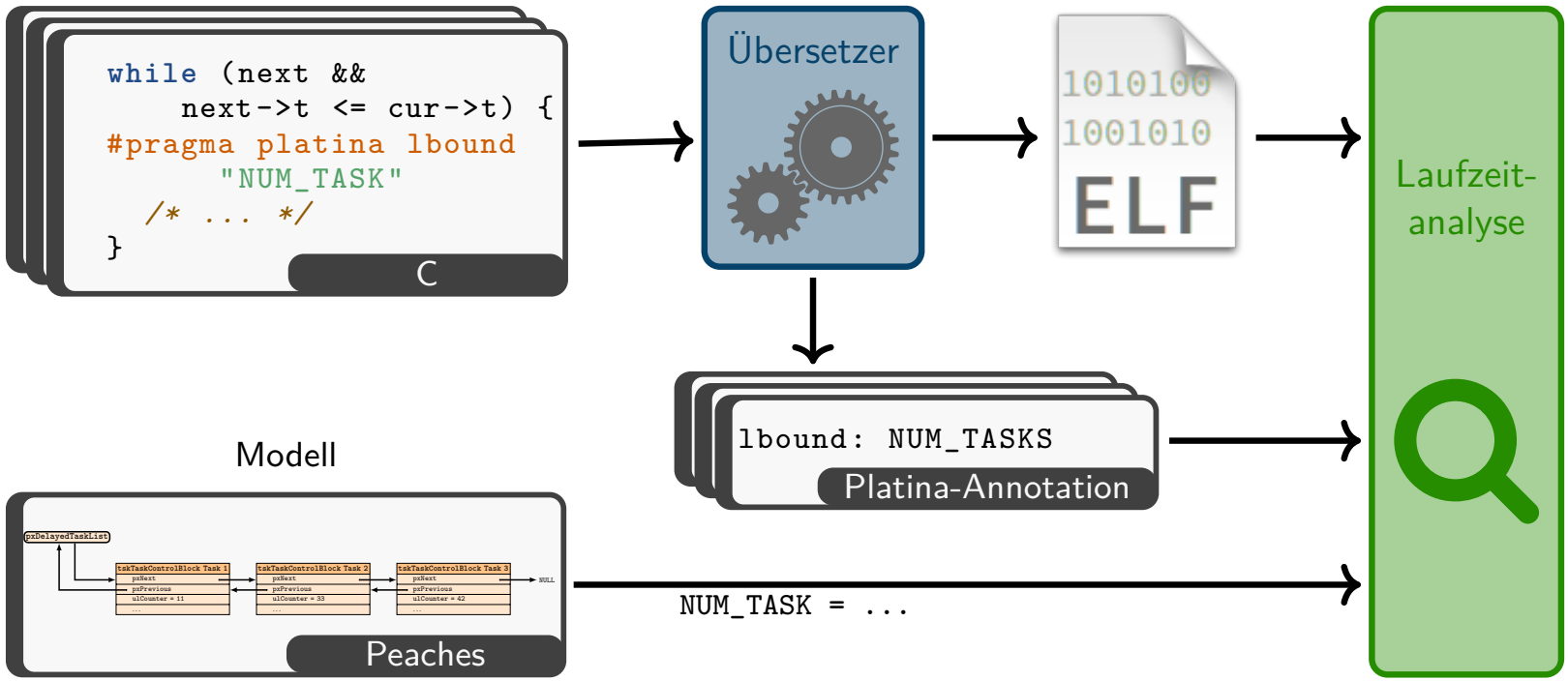
Architektur



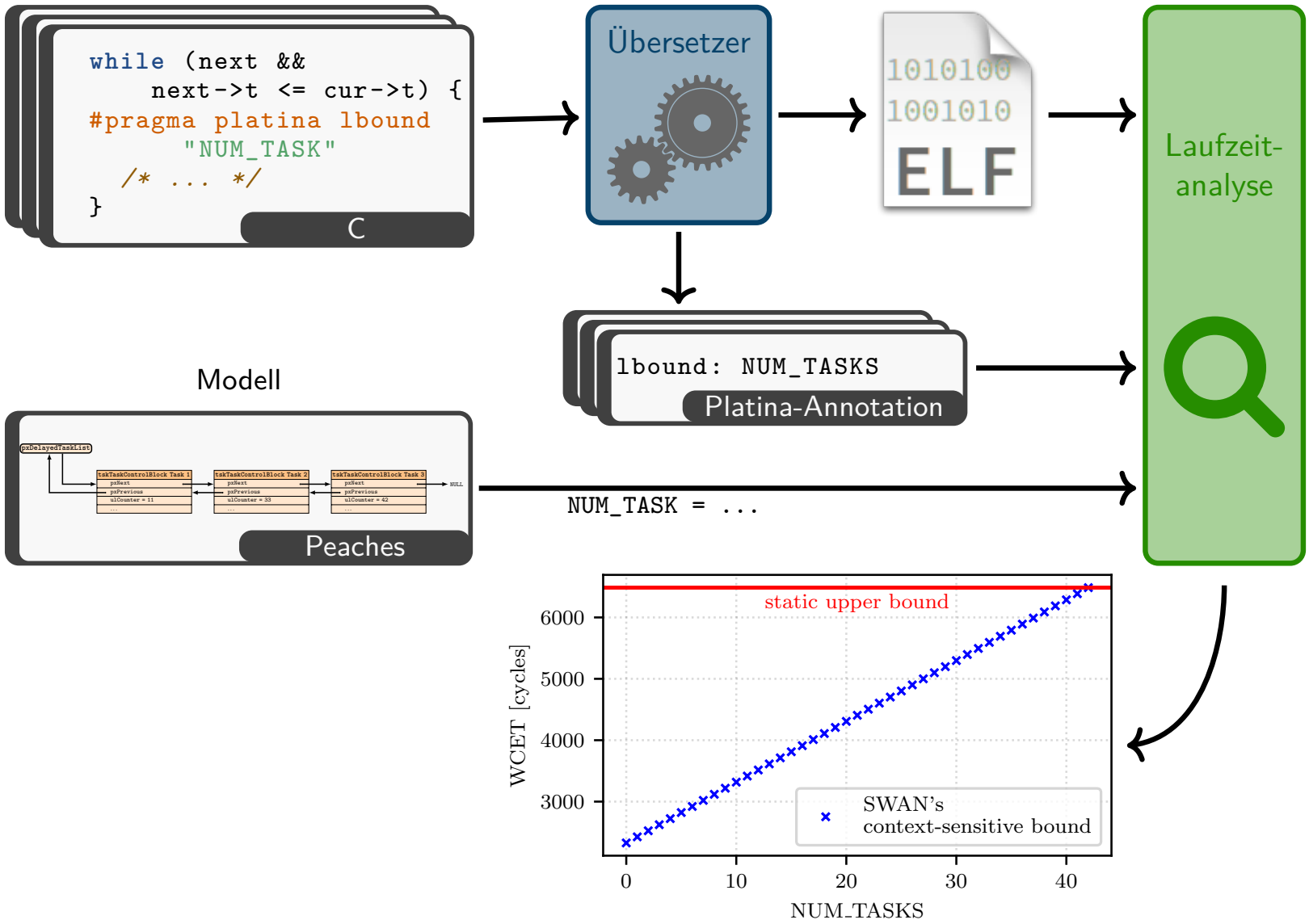
Architektur



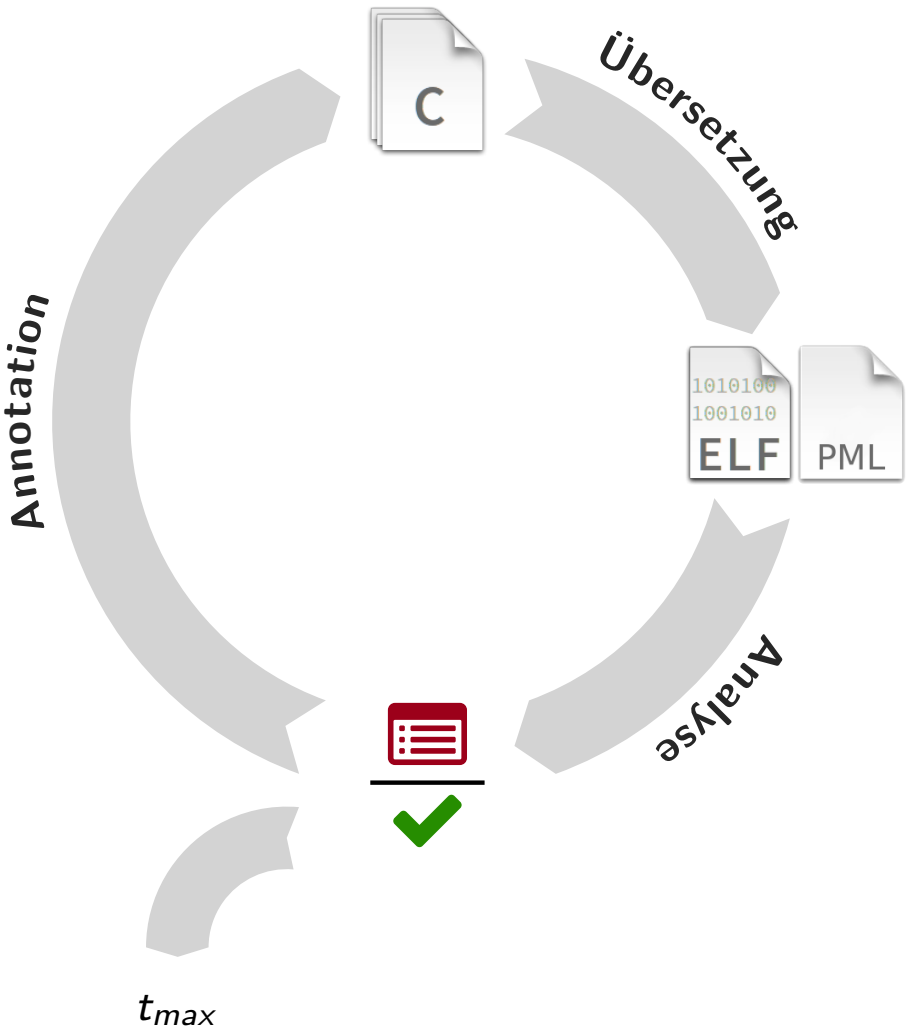
Architektur



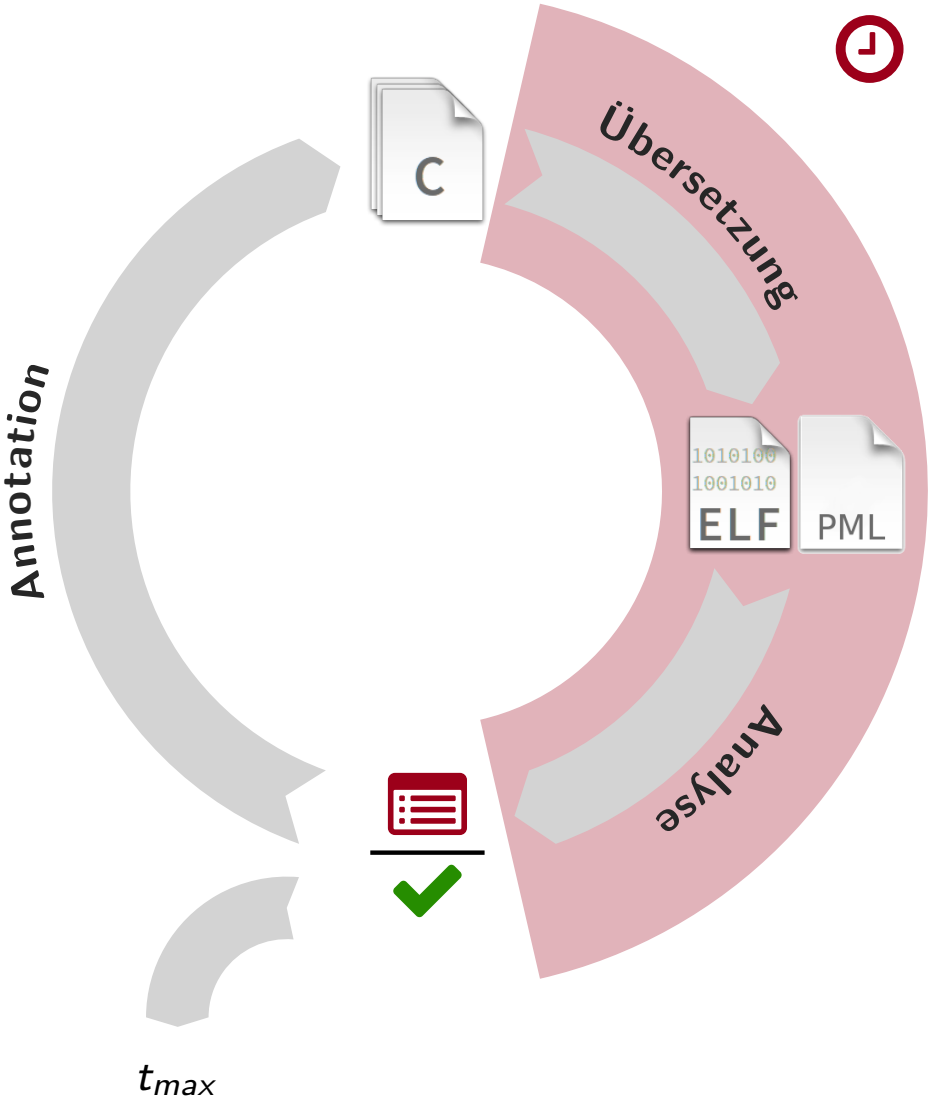
Architektur



Arbeitsablauf

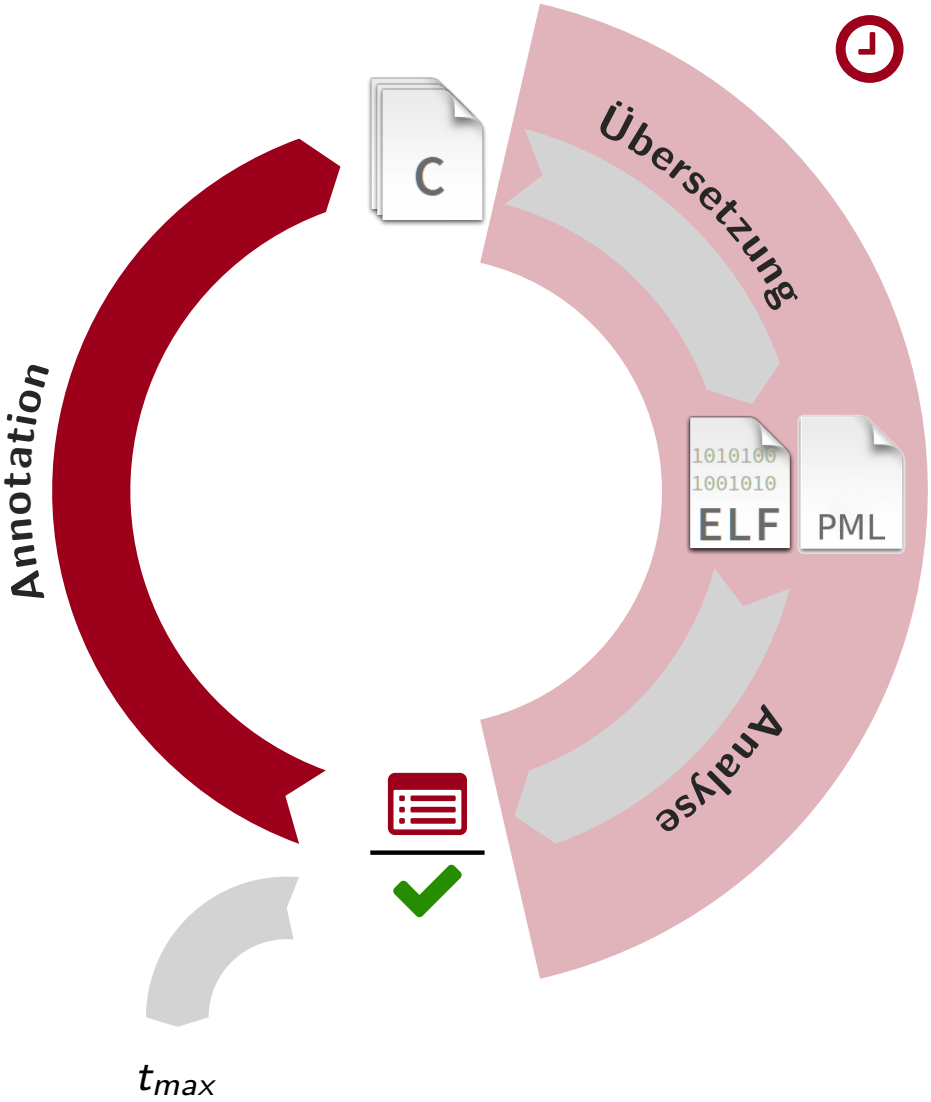


Arbeitsablauf



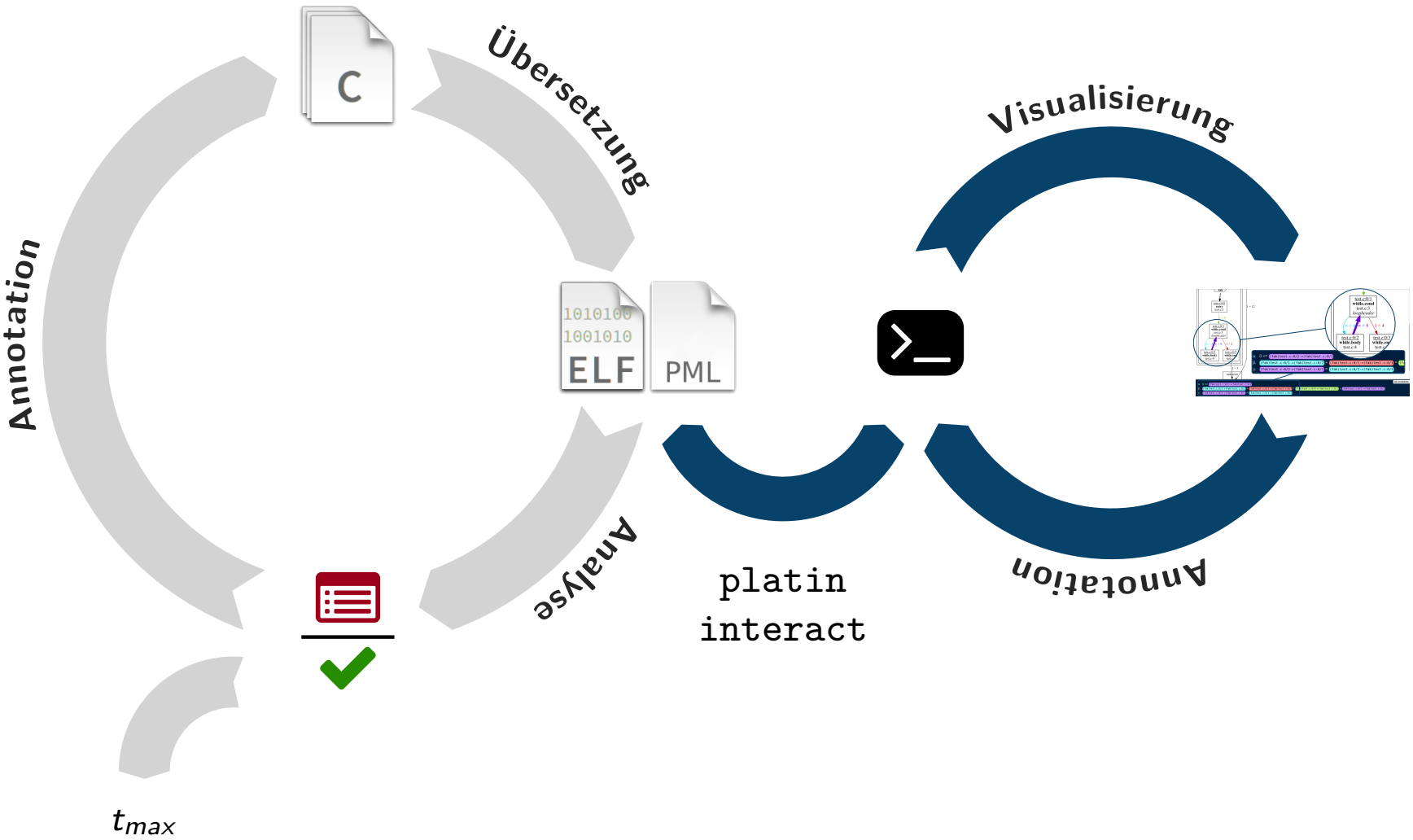
Linuxkern: über 15 min
40 Kerne @ 2.20 GHz
(Xeon E5-4640)
128GB RAM

Arbeitsablauf



Linuxkern: über 15 min
40 Kerne @ 2.20 GHz
(Xeon E5-4640)
128GB RAM

Arbeitsablauf



Interaktive Annotation

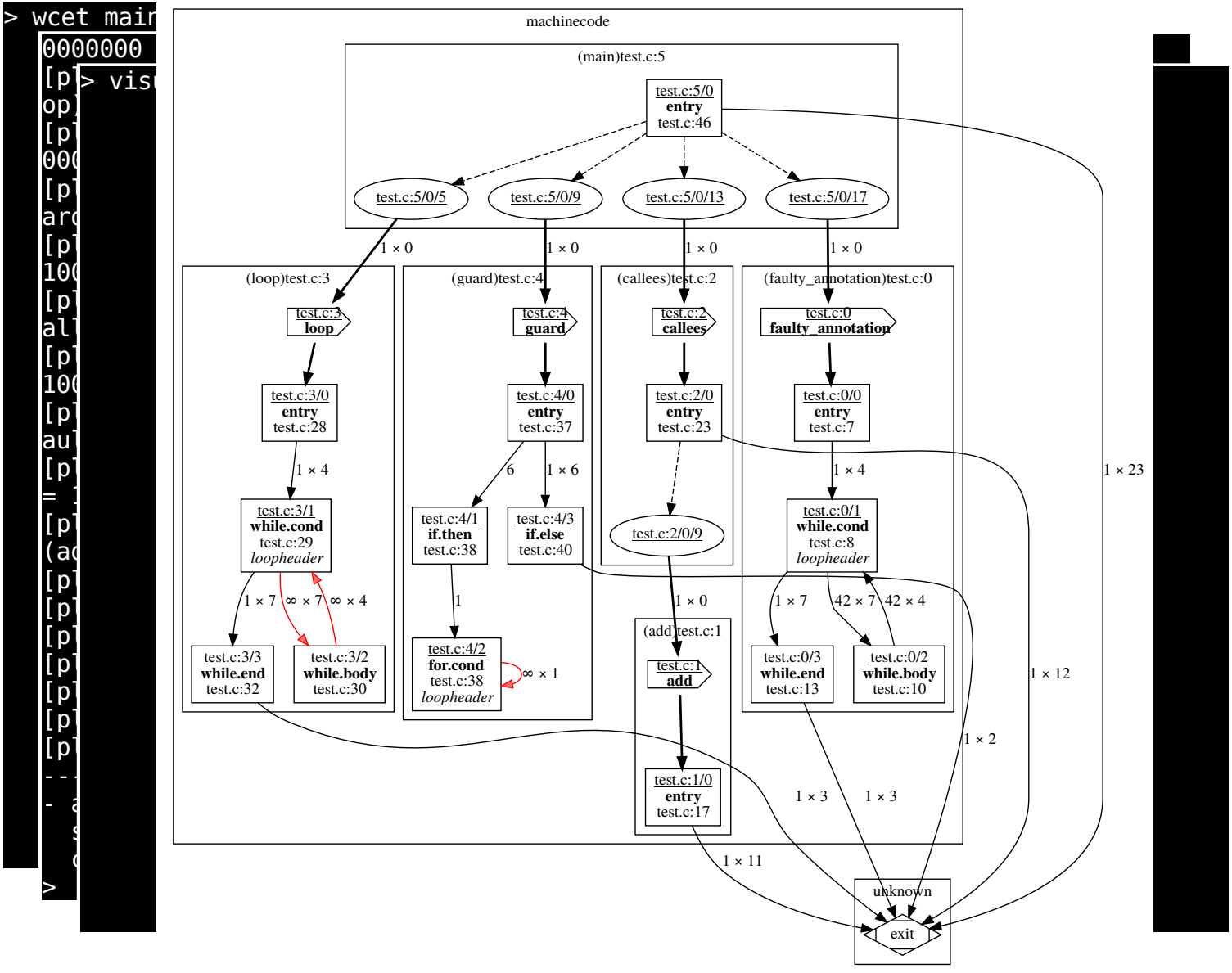
```
> wcet main
```

Interaktive Annotation

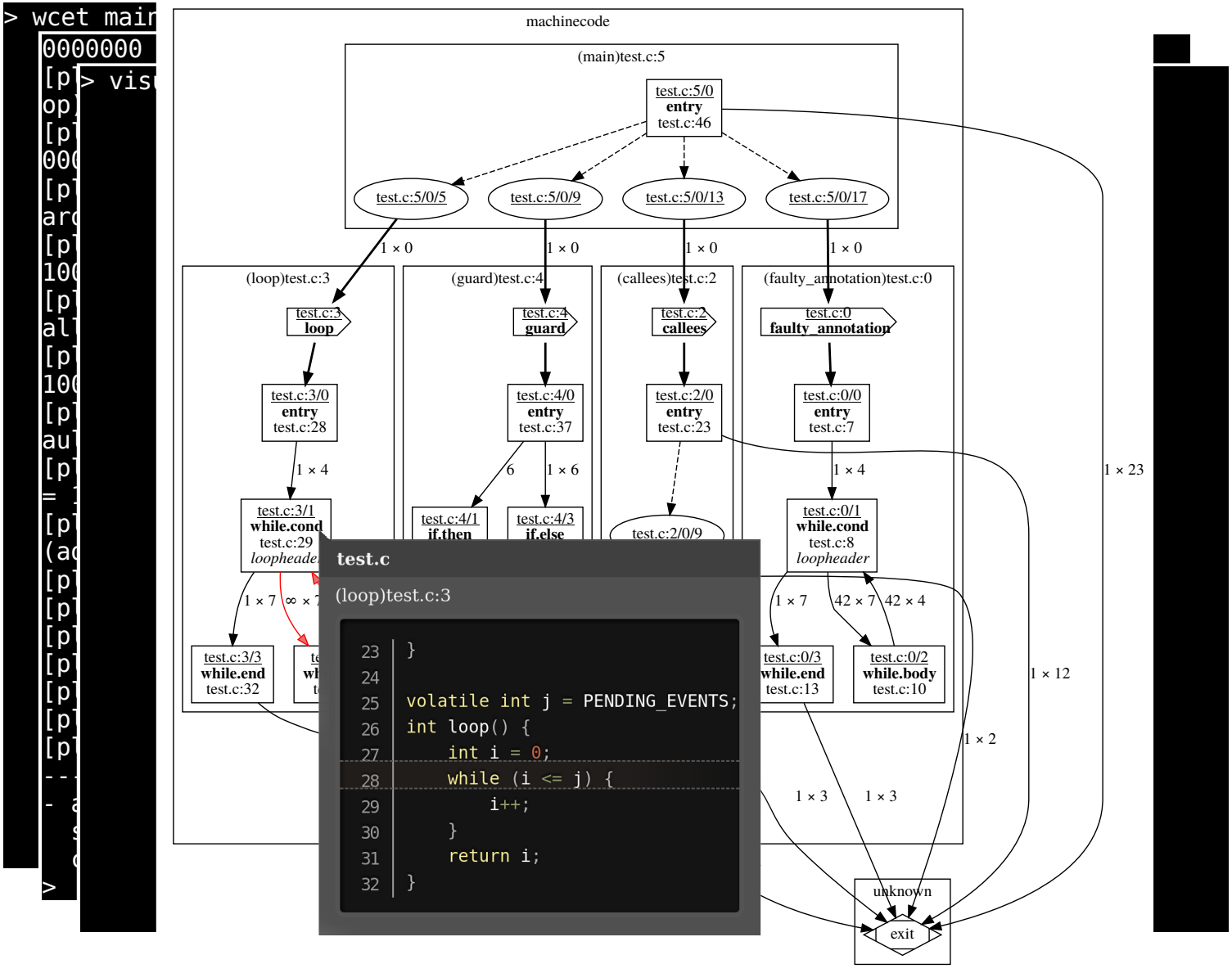
```
> wcet main
0000000
[platin] DEBUG:  74: constraint __debug_upper_bound_v20: (main)test.c:5/0/5->(lo
op)test.c:3 <= 10000000
[platin] DEBUG:  75: constraint __debug_upper_bound_v21: (main)test.c:5/0/9 <= 1
0000000
[platin] DEBUG:  76: constraint __debug_upper_bound_v22: (main)test.c:5/0/9->(gu
ard)test.c:4 <= 10000000
[platin] DEBUG:  77: constraint __debug_upper_bound_v23: (main)test.c:5/0/13 <=
10000000
[platin] DEBUG:  78: constraint __debug_upper_bound_v24: (main)test.c:5/0/13->(c
allees)test.c:2 <= 10000000
[platin] DEBUG:  79: constraint __debug_upper_bound_v25: (main)test.c:5/0/17 <=
10000000
[platin] DEBUG:  80: constraint __debug_upper_bound_v26: (main)test.c:5/0/17->(f
aulty_annotation)test.c:0 <= 10000000
[platin] DEBUG:  81: constraint __debug_upper_bound_v27: (callees)test.c:2/0/9 <
= 10000000
[platin] DEBUG:  82: constraint __debug_upper_bound_v28: (callees)test.c:2/0/9->
(add)test.c:1 <= 10000000
[platin] DEBUG: #<Set:0x00000002733168>
[platin] WARNING: ILP: Unbounded loops: (guard)test.c:4/2, (loop)test.c:3/1
[platin] WARNING: Unbounded hint [(guard)test.c:4/2]: test.c:38
[platin] WARNING: Unbounded hint [(loop)test.c:3/1]: test.c:29
[platin] WARNING: WCA: ILP failed: LPSolver Error: UNBOUNDED (E3)
[platin] INFO: Finished run WCET analysis (platin)          in 19 ms
[platin] INFO: best WCET bound: -1 cycles
---
- analysis-entry: main
  source: platin
  cycles: -1
>
```

Interaktive Annotation

Interaktive Annotation

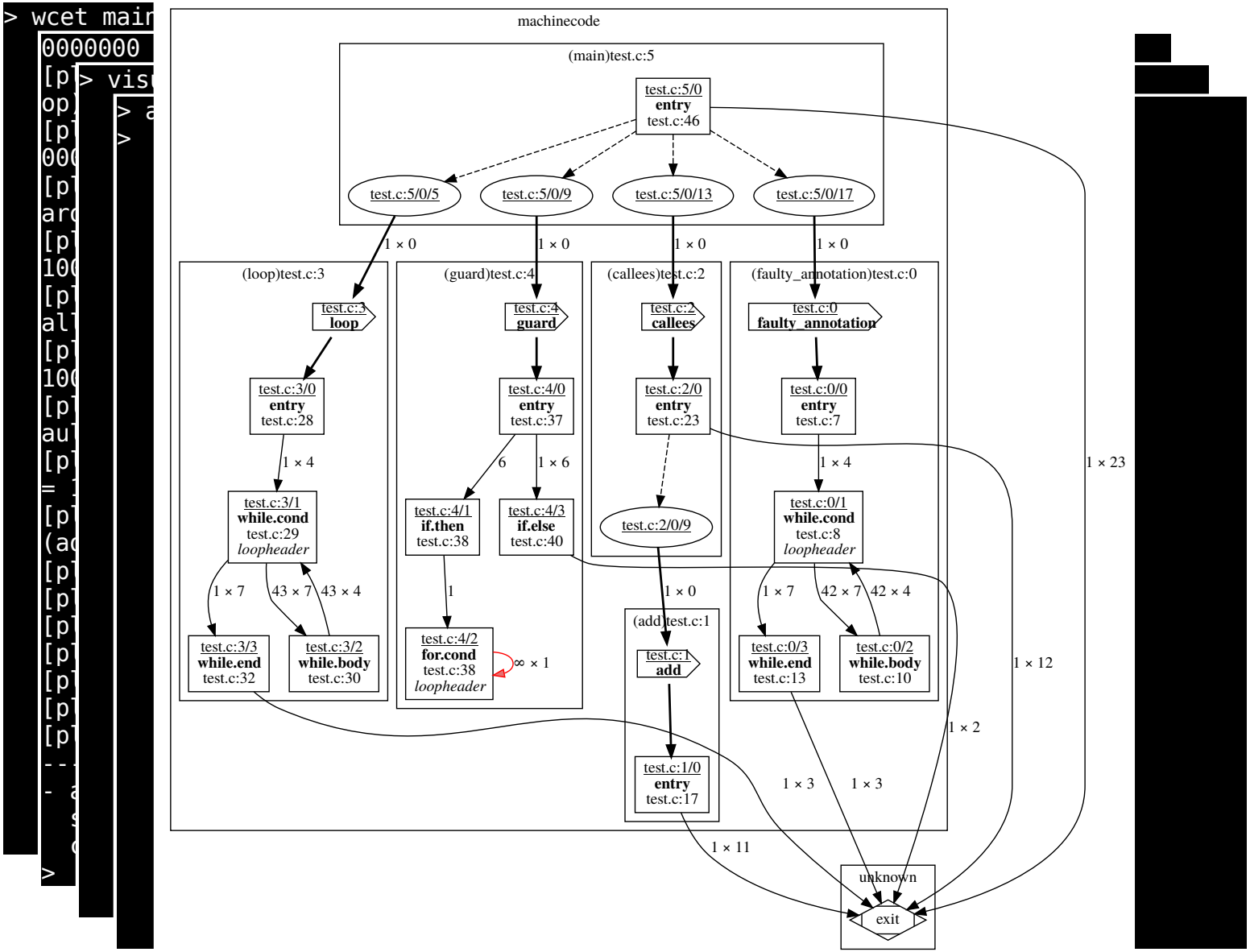


Interaktive Annotation



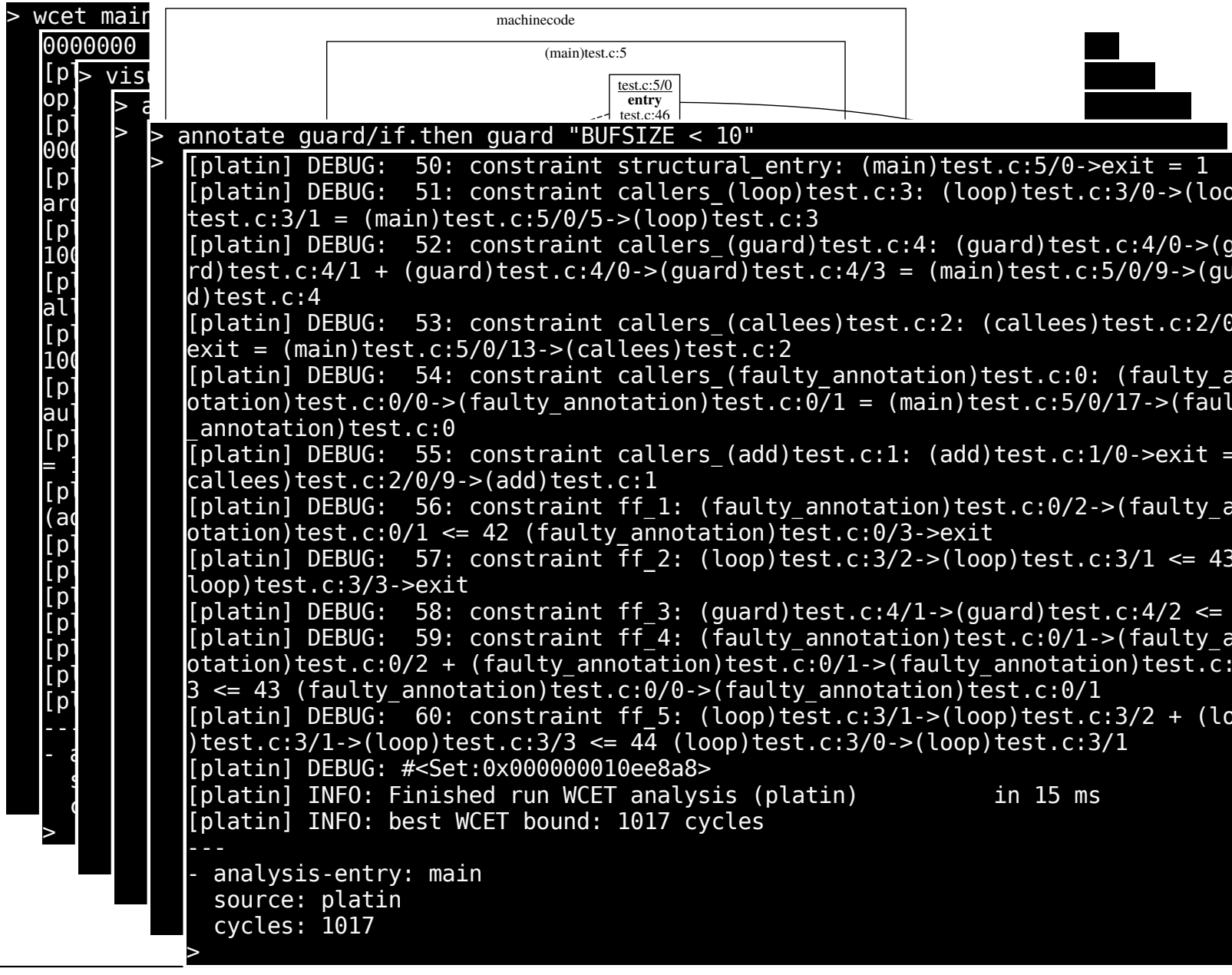
Interaktive Annotation

Interaktive Annotation

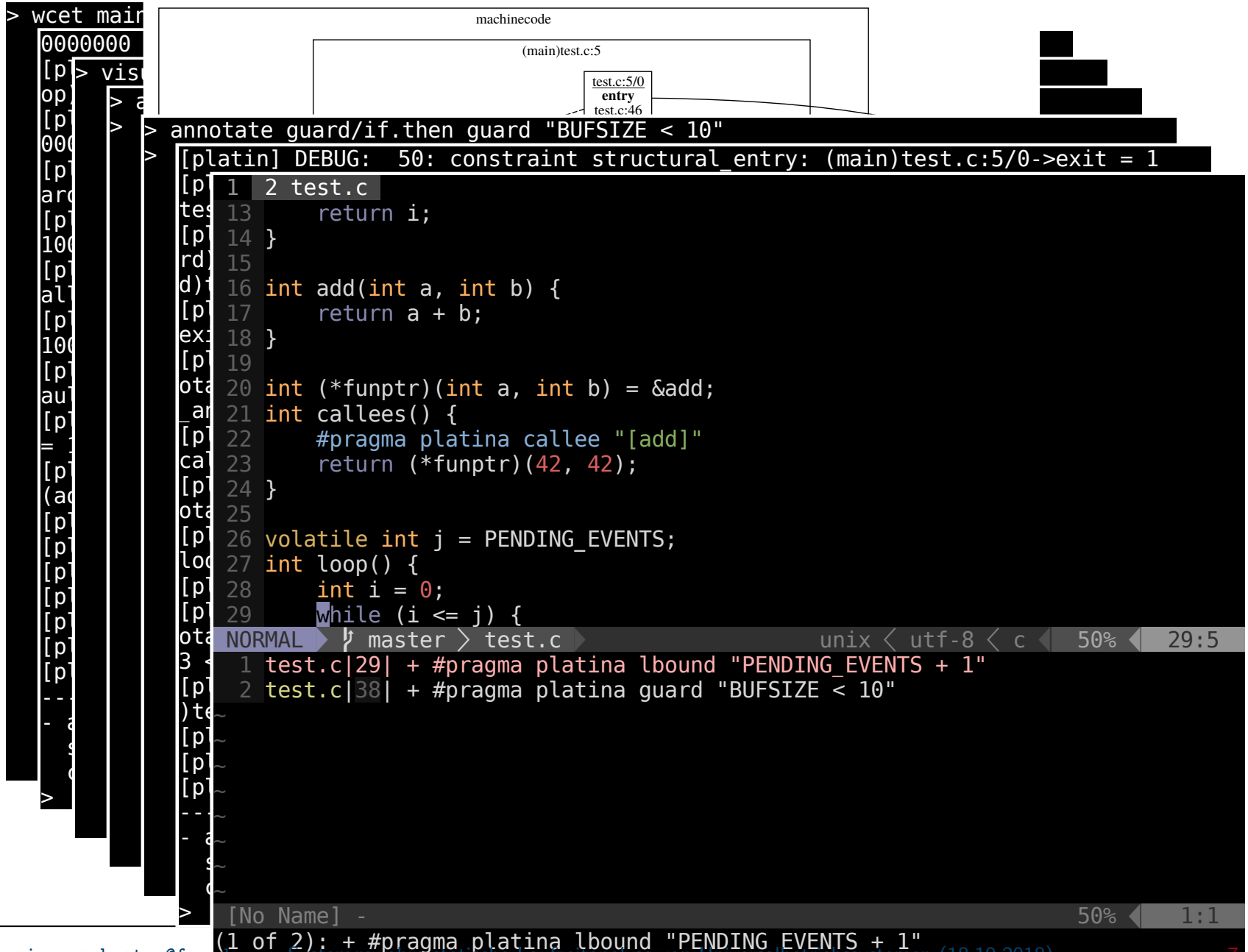


Interaktive Annotation

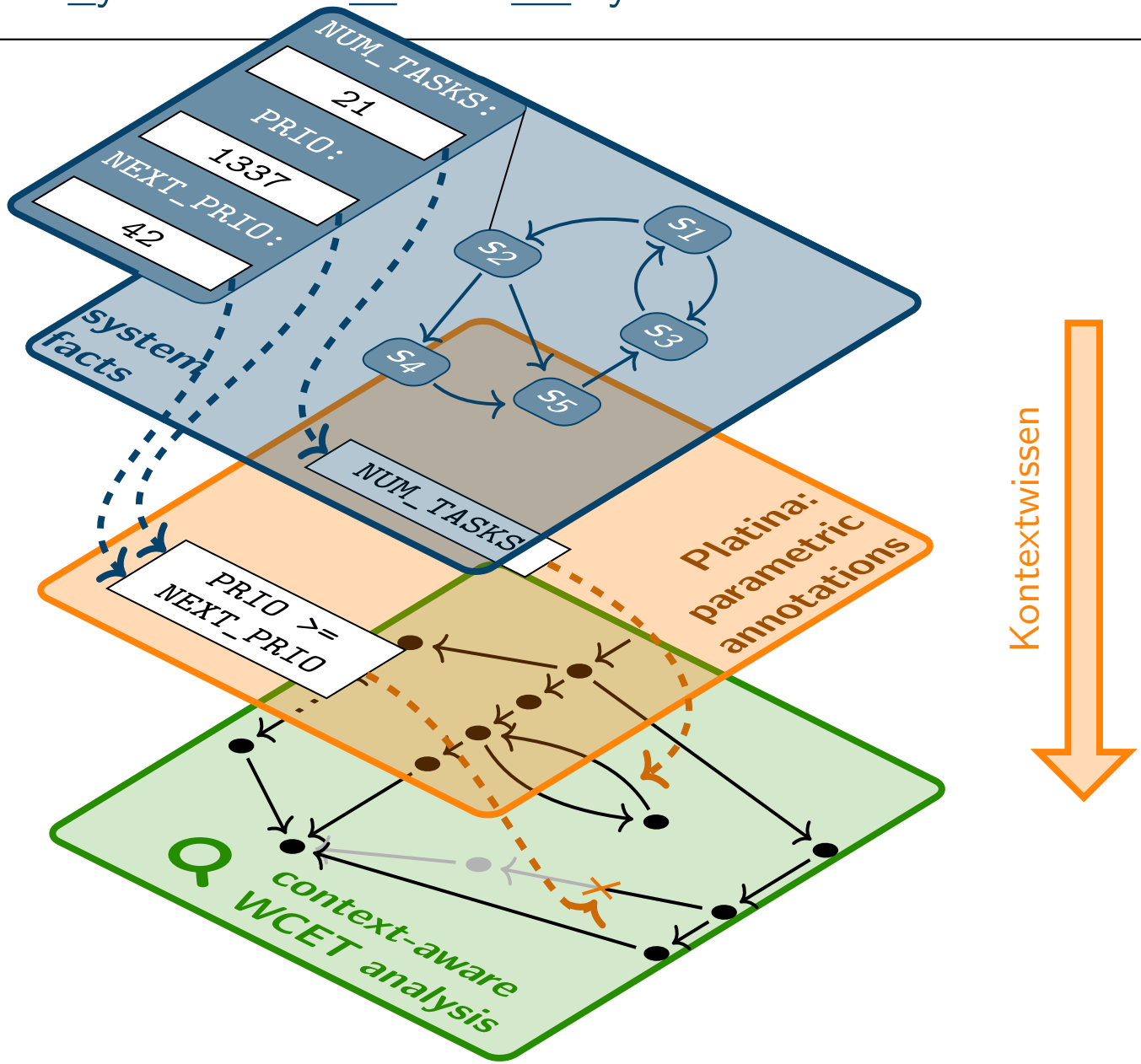
Interaktive Annotation



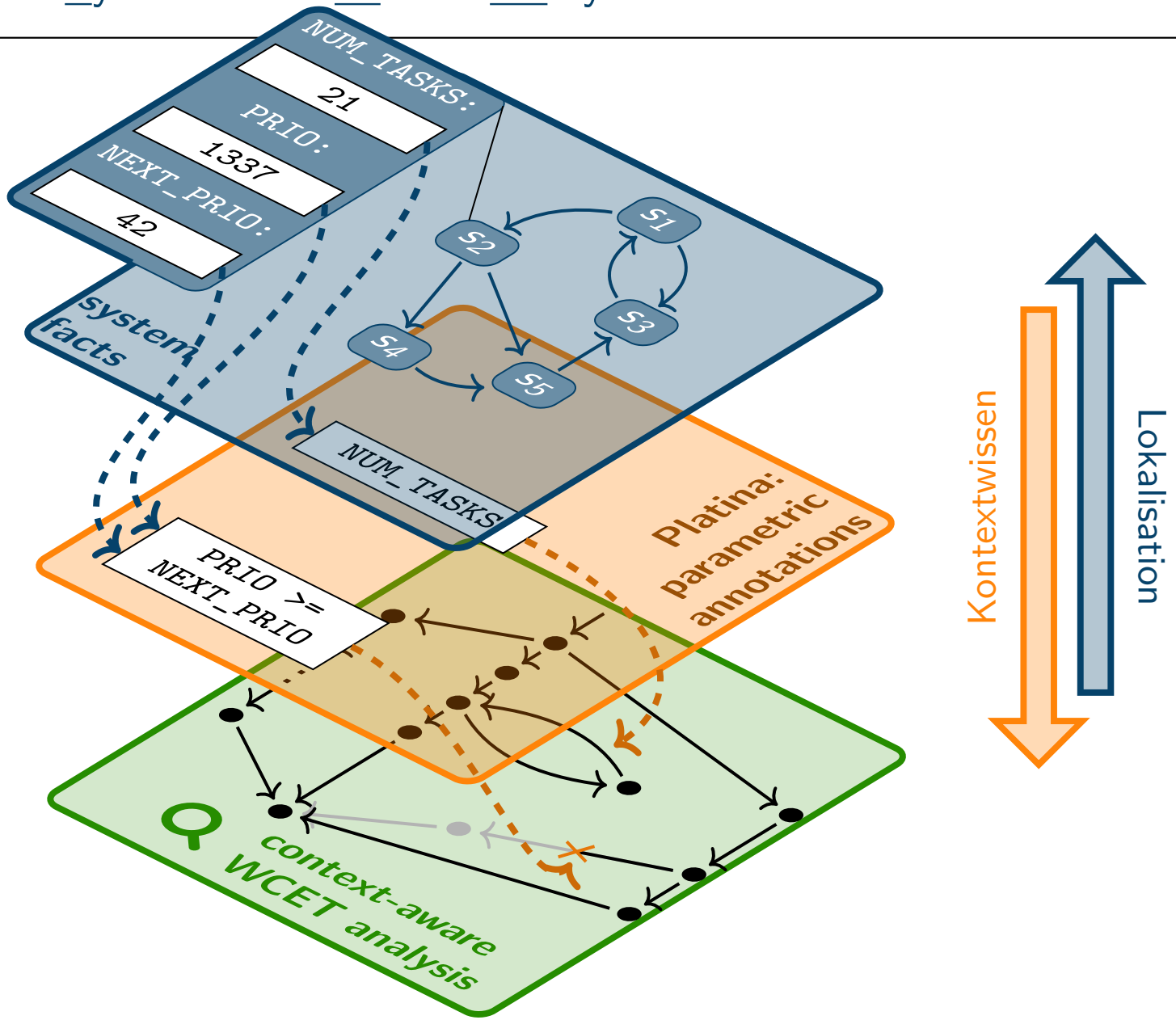
Interaktive Annotation



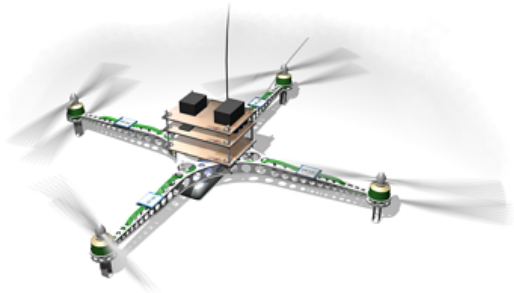
SWAN: System-wide WCET Analyzer

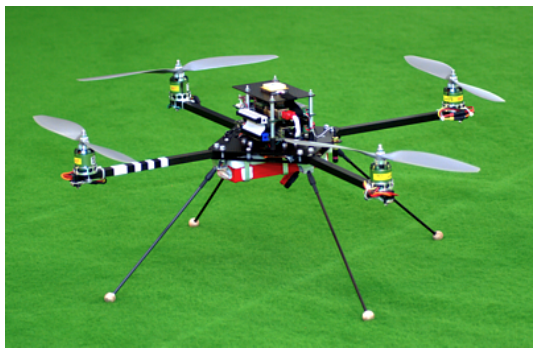


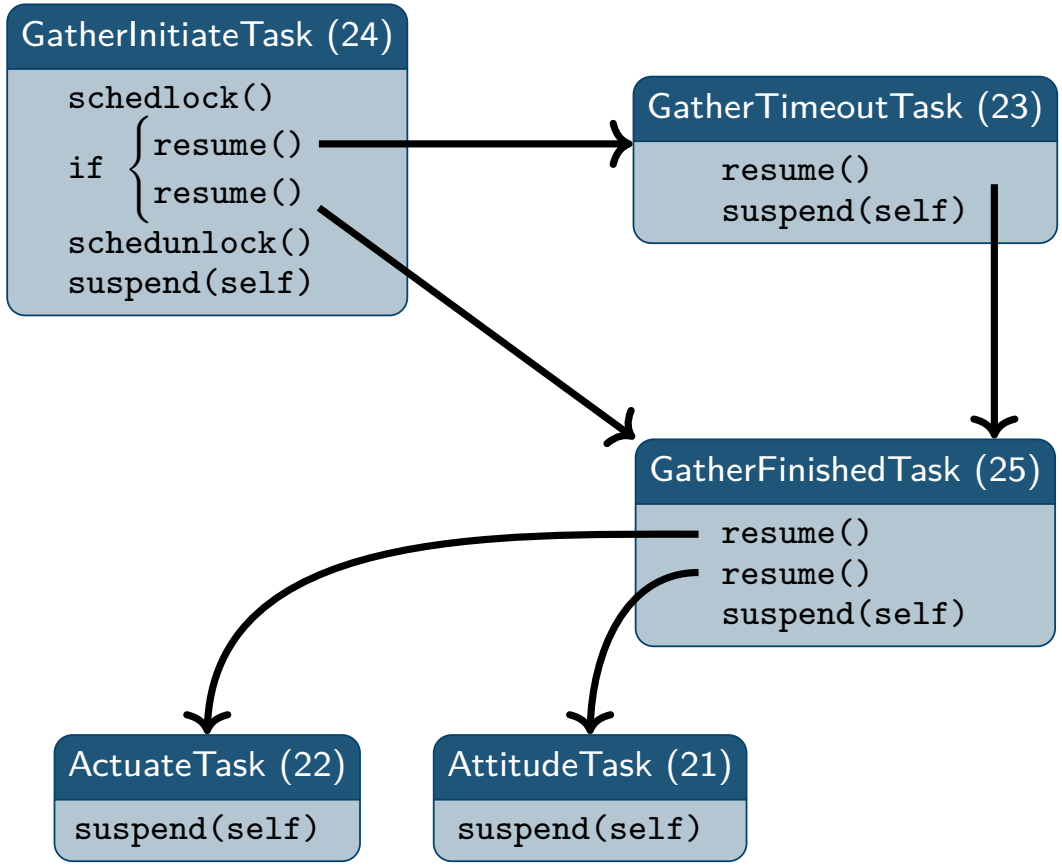
SWAN: System-wide WCET Analyzer



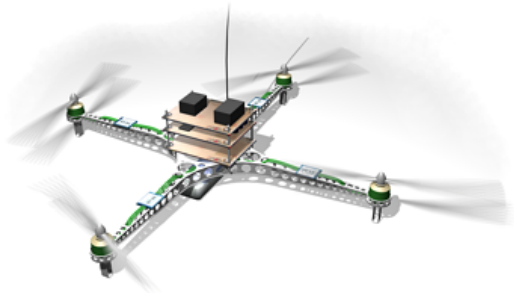
Fallstudie I4Copter (I)

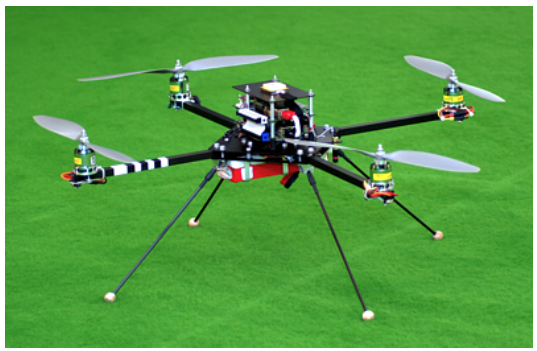


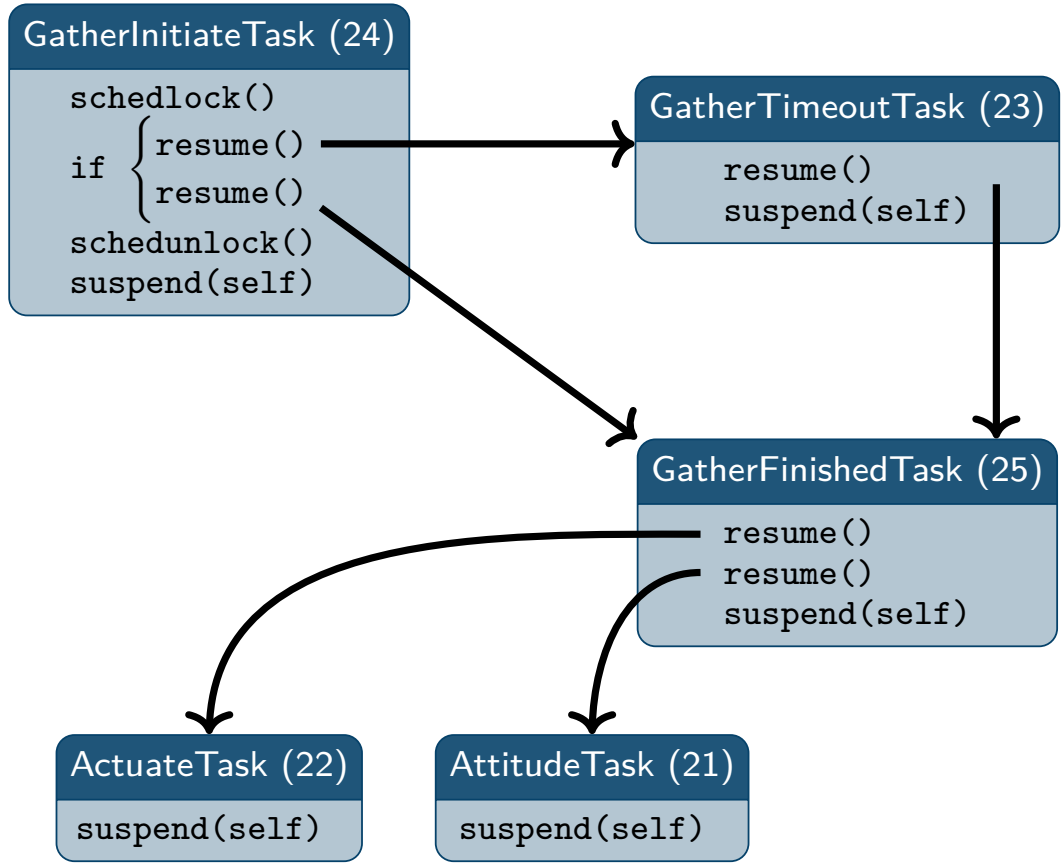




Fallstudie I4Copter (I)

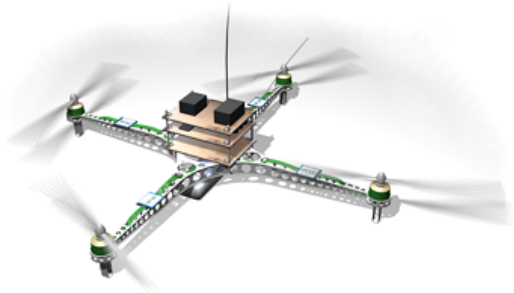


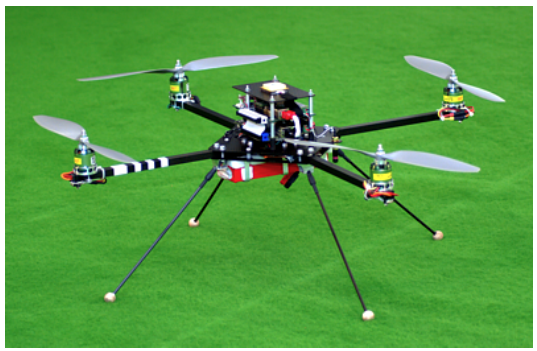


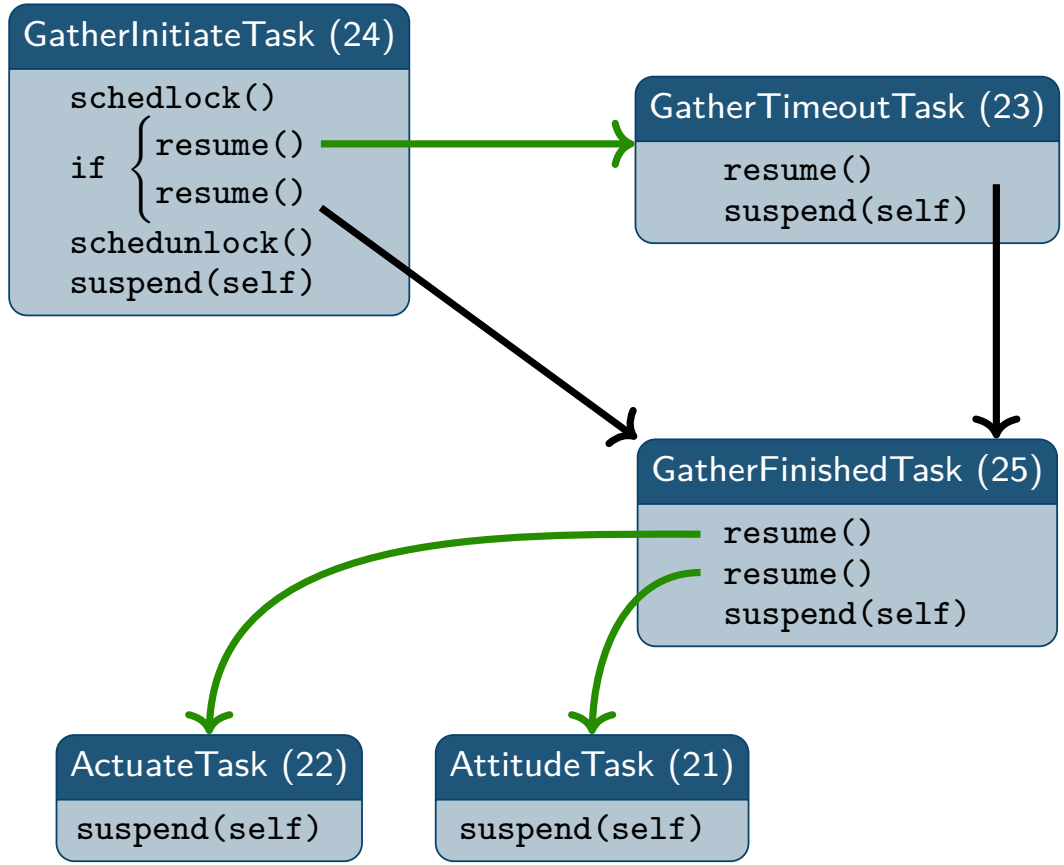



```
void ResumeTask(struct tcb* next) {
  // ...
  if (next->prio > cur->prio) {
  #pragma platina guard "NEXT_PRIO > CUR_PRIO"
    reschedule(); // expensive
  } // ...
}
```

Fallstudie I4Copter (I)

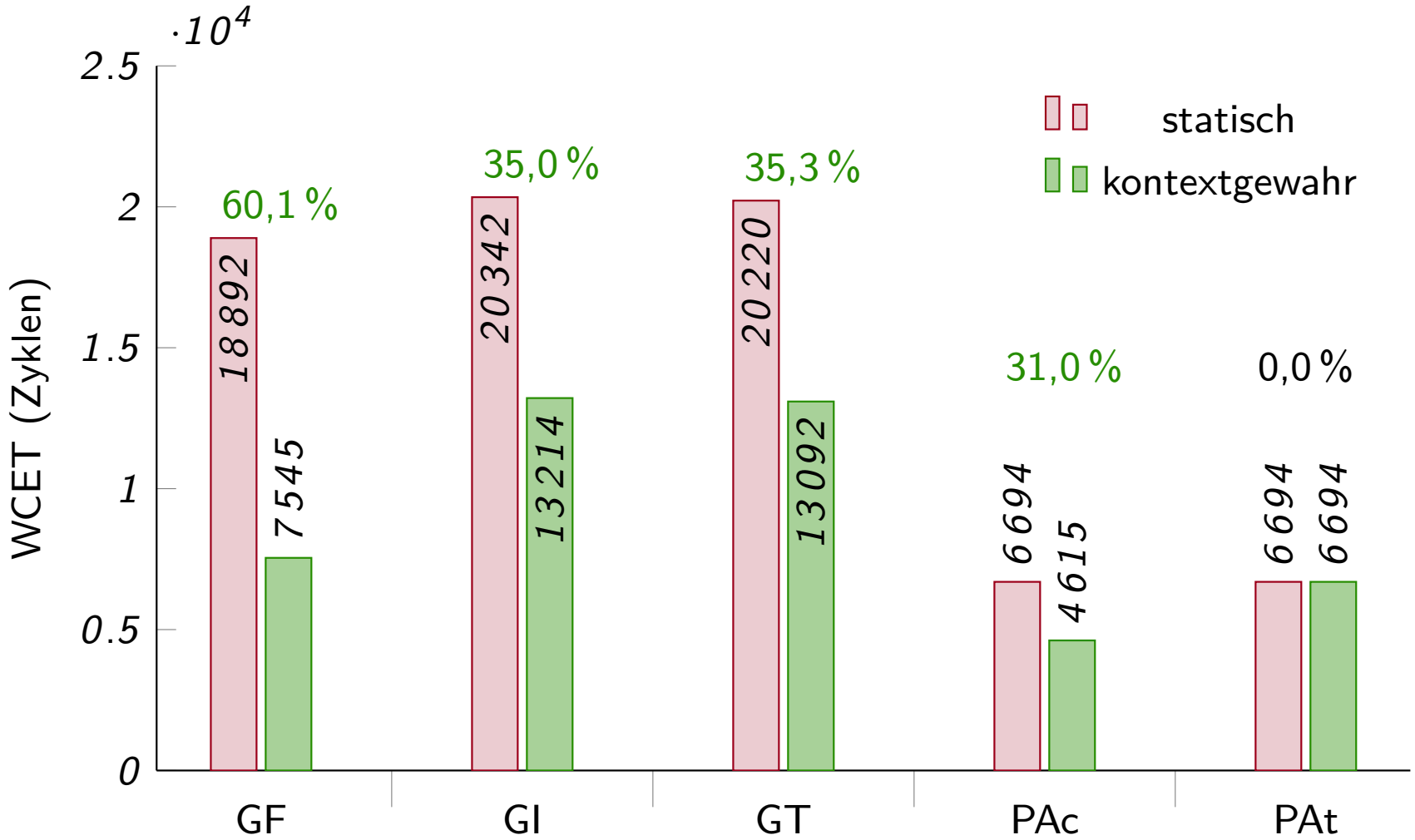




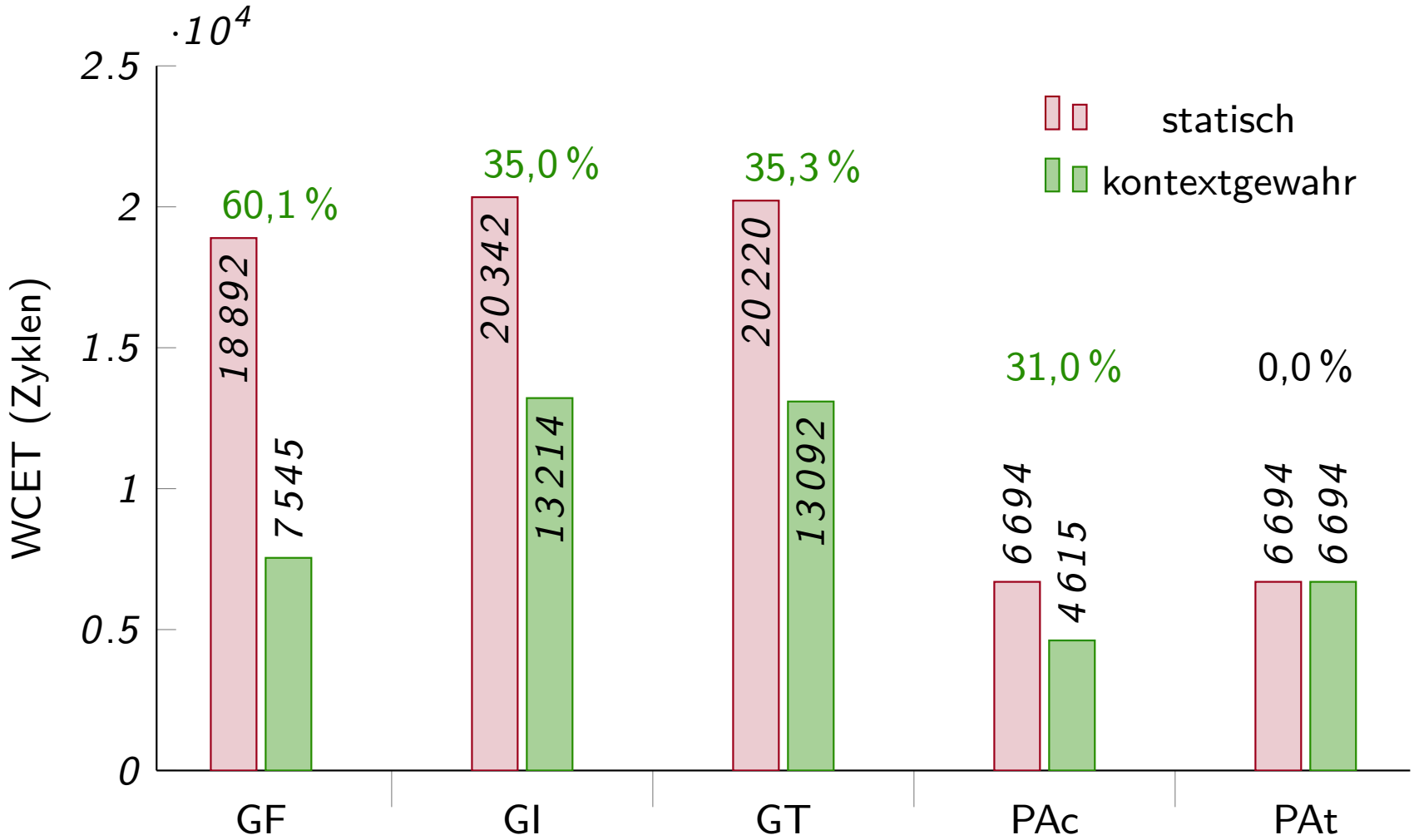



```
void ResumeTask(struct tcb* next) {
  // ...
  if (next->prio > cur->prio) {
  #pragma platina guard "NEXT_PRIO > CUR_PRIO"
    reschedule(); // expensive
  } // ...
}
```

Fallstudie I4Copter (II)



Fallstudie I4Copter (II)



⇒ Schlimmstmögliche Antwortzeit des Aufgabensystems um **40,7 %** geringer

Zusammenfassung

- statische Laufzeitanalyse von Universalbetriebssystemen problematisch
  - Pfadidentifikation
  - Abhängigkeit vom dynamischen Laufzeitkontext

⇒ Lösungsansatz: modellbasierte, parametrierbare statische Laufzeitanalyse

✓ Annotationssprache auf Hochsprachenebene
✓ Optimierungsgewahre Integration in Übersetzer und Laufzeitanalyse
✓ Effiziente Annotation durch interaktive, integrierte Annotationsumgebung

- Ermöglicht statische Laufzeitanalyse kompletter Betriebssysteme

⇒ Kontextgewahre Analyse ermöglicht über 40 % bessere Schranken

`https://gitlab.cs.fau.de/SWAN/`