

# Echtzeitverhalten durch die Verwendung von CPU Stubs: Eine Erweiterung von Dynamic Performance Stubs

**Echtzeit 2009**



1

**Einleitung**

2

**Dynamic Performance Stubs (Übersicht)**

3

**Dynamic Performance Stubs (Framework)**

4

**Methodik CPU Stubs**

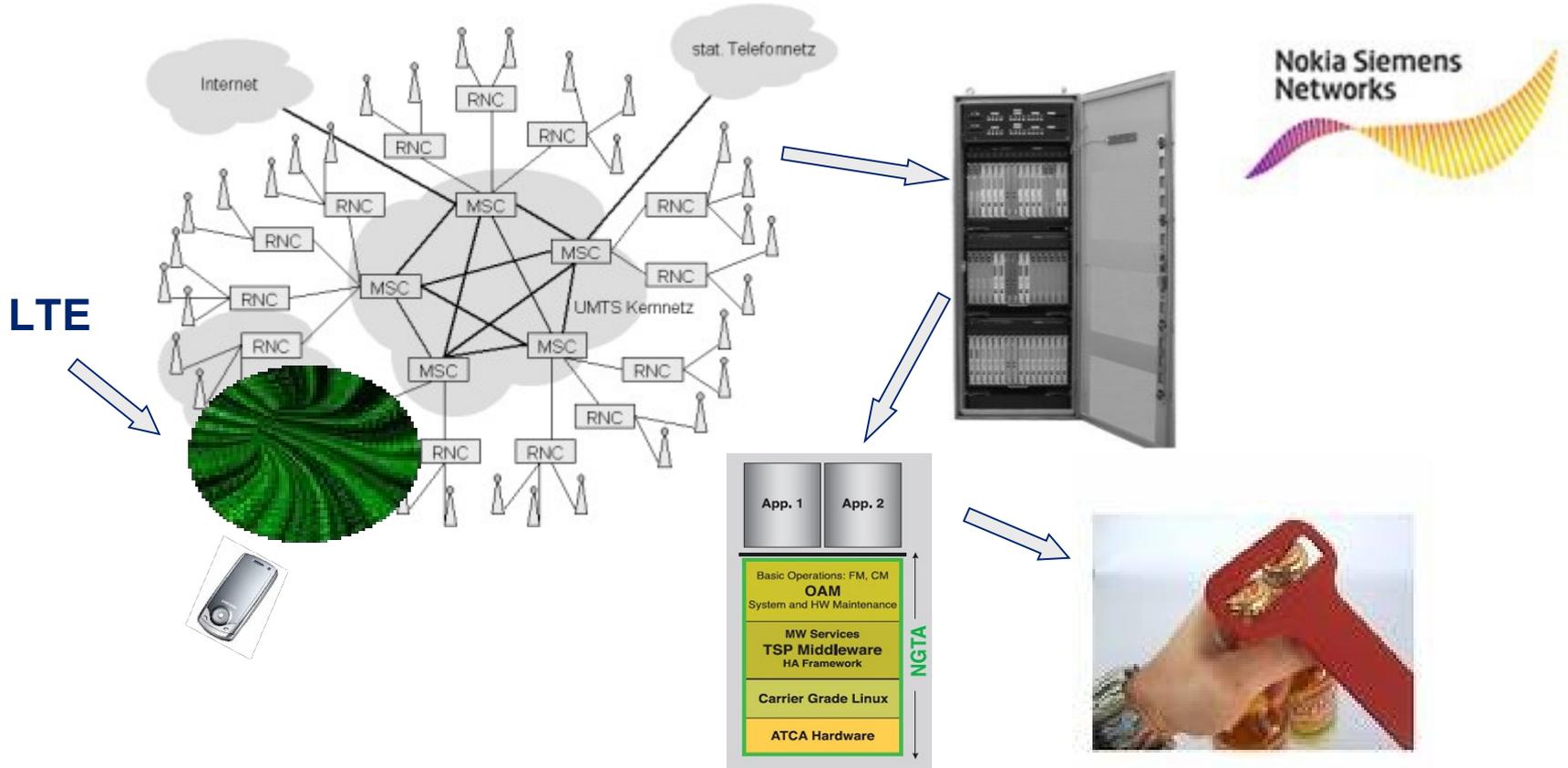
5

**Fallstudie CPU Stubs**

6

**Zusammenfassung & Weiteres Vorgehen**

# Software Performance Optimierung in LTE Telekommunikationssystemen



Welche Probleme gibt es bei Performance Optimierungen?



## Welche Probleme gibt es bei Performance Optimierungen?



- **Generell**

- **Optimierungen werden meist erst sehr “spät”  
druchgeführt**
- **Verursachen häufig hohen Aufwand, welcher nicht  
kalkuliert wurde**
- **Abschätzungen sind meist zu ungenau**

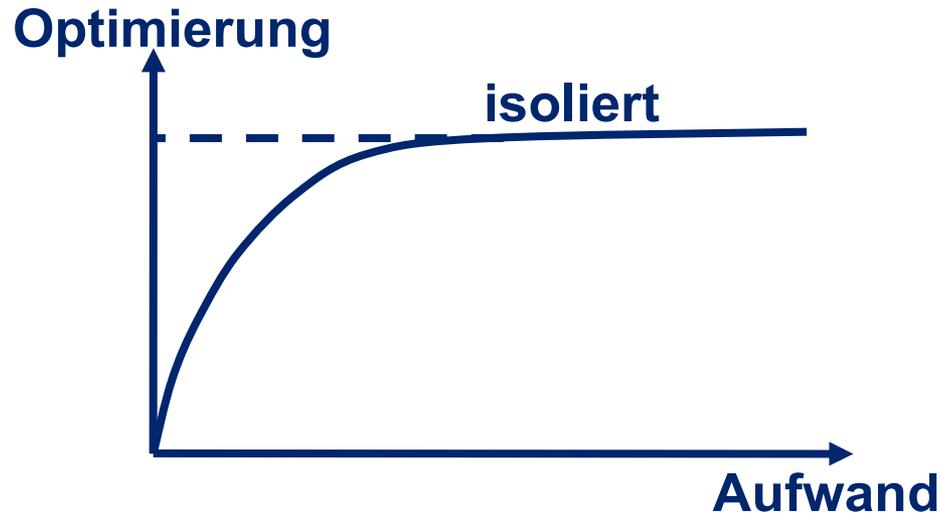
## Probleme bei der Optimierung I

---

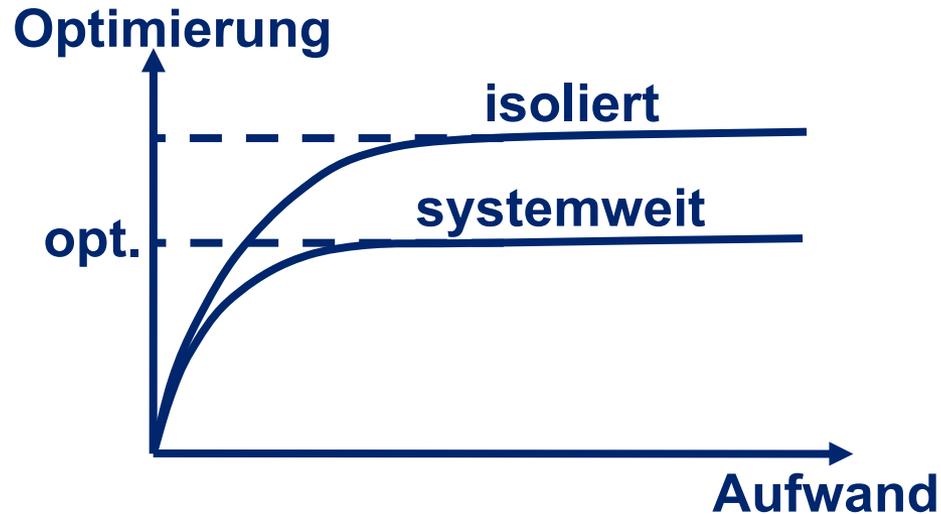
### Welche Probleme gibt es bei Performance Optimierungen?

- **Technisch**
  - **Lediglich der nächste Falschenhals ist sichtbar**
  - **Verbesserungen haben häufig nur geringe Auswirkungen**
  - **Optimierungen hängen voneinander ab**
  - **Der Quellcode wird häufig schwer verständlich, aufwendig zu warten und zu debuggen**
  - **Während Wartungsarbeiten können Optimierungen unbeabsichtigt rückgängig gemacht werden**

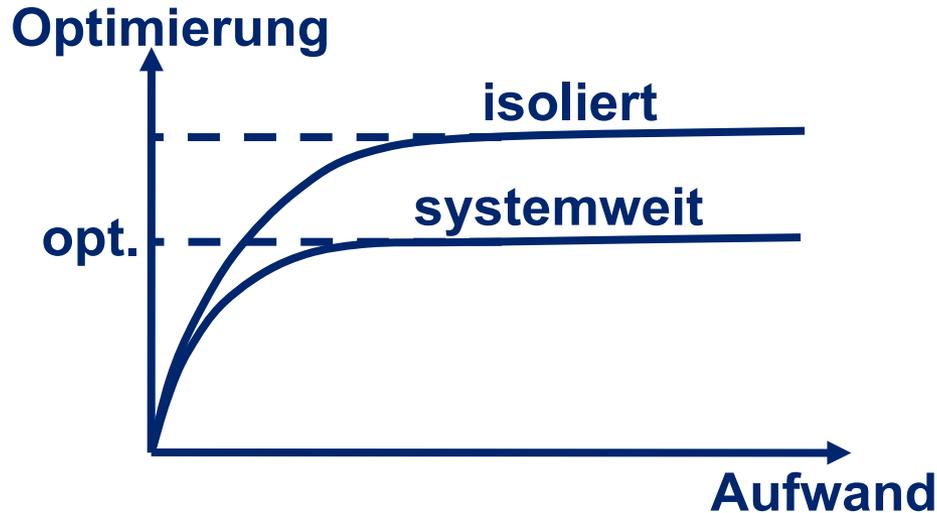
## Optimierungsfunktion eines einzelnen Software Moduls (vereinfacht)



## Optimierungsfunktion eines SW Moduls innerhalb des Systems (vereinfacht)



## Optimierungsfunktion eines SW Moduls innerhalb des Systems (vereinfacht)

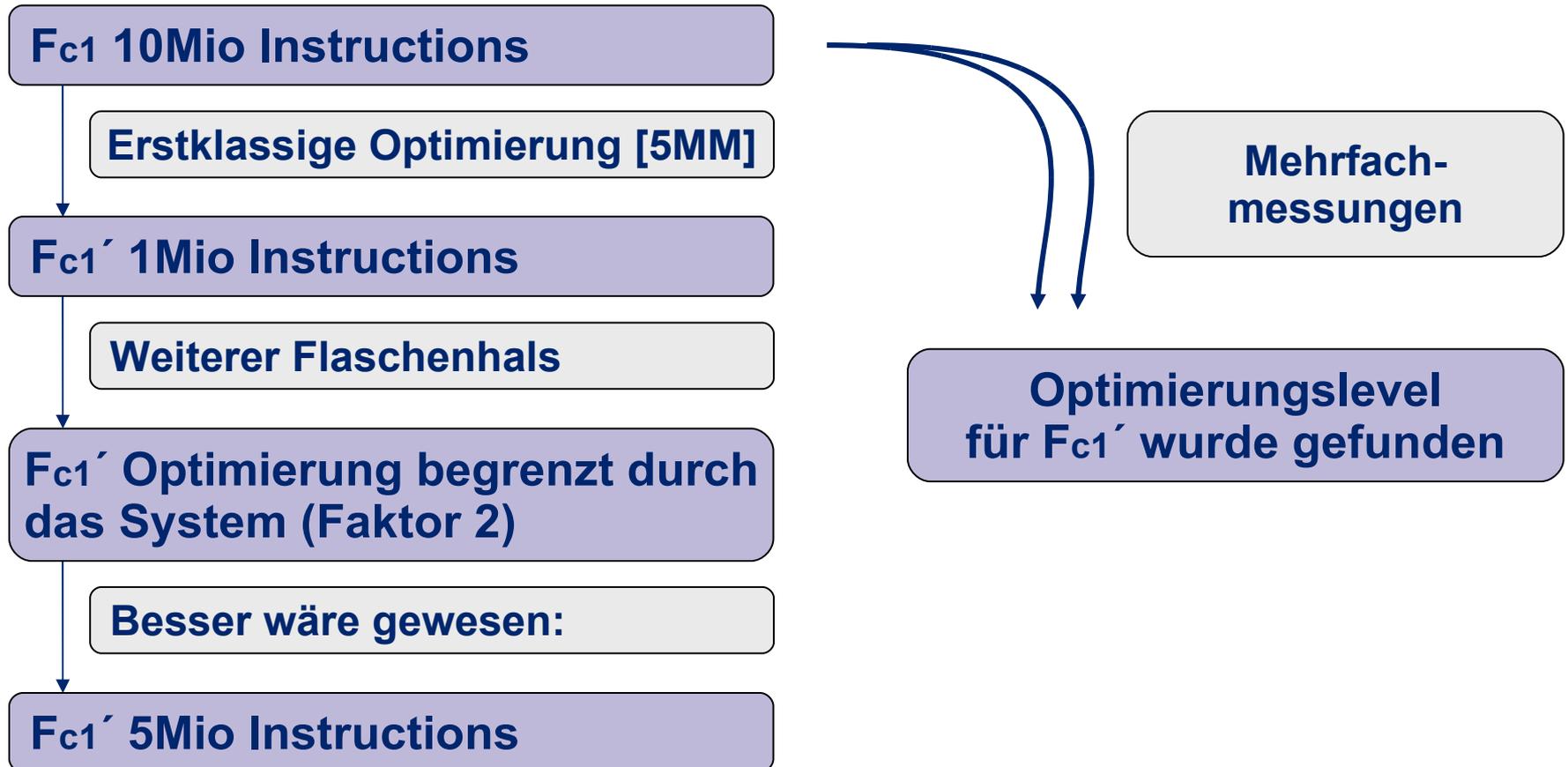


→ **Wie kann das Optimum  
gefunden werden?**

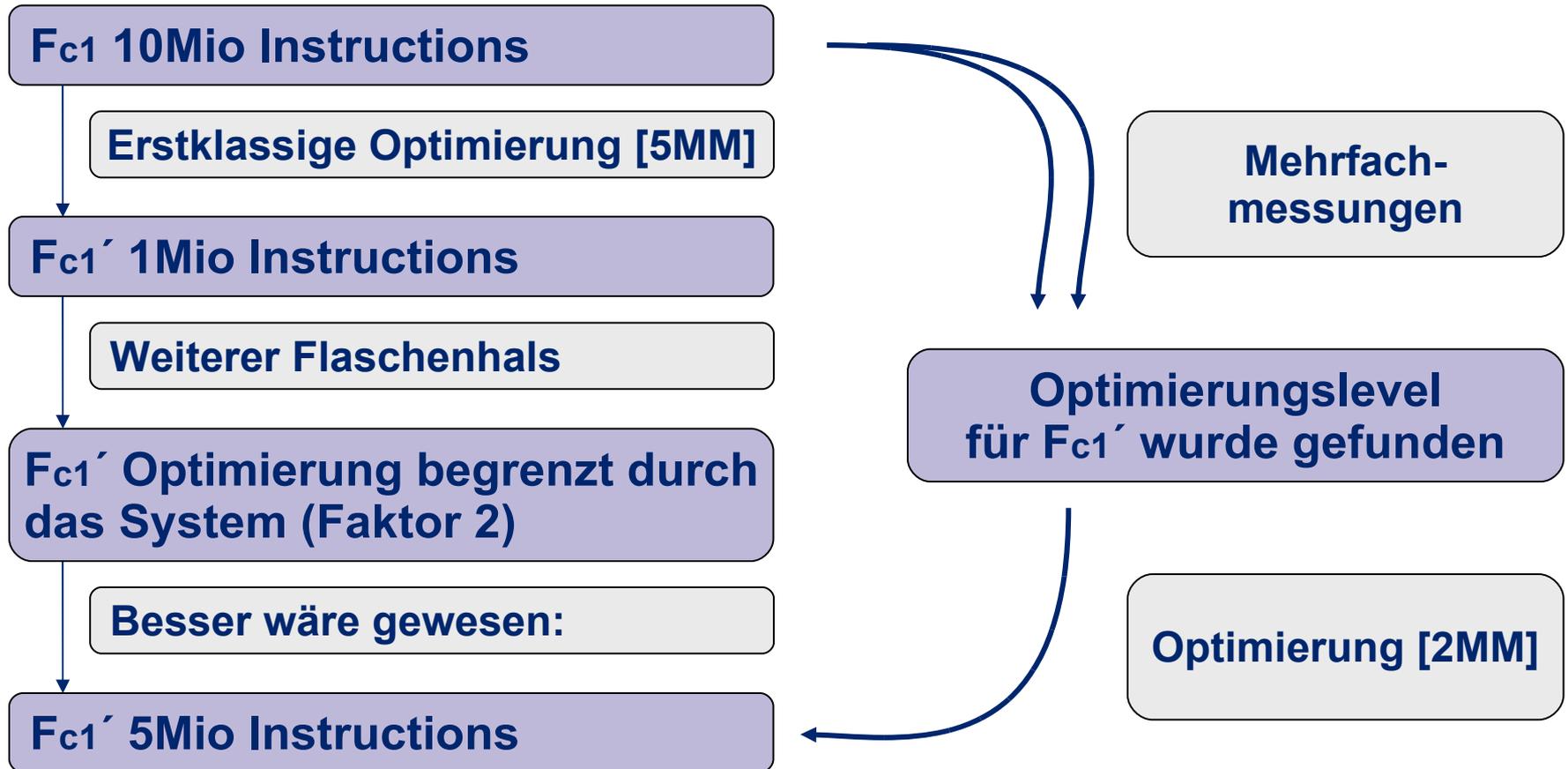
## Wie wird Software optimiert?



## Wie wird Software optimiert?



## Wie wird Software optimiert?



### Was sind die Vorteile?

- Minimierung des Aufwands bei maximalen Gewinn
  - Aufwand kann besser geschätzt werden
  - Fundierte Abschätzungen des Gewinns
- } Zielorientierte Optimierung
- „Load & Stress“ Tests sind verfügbar (bei geringem Aufwand)
  - „Idealisierte“ Funktionen können gemessen werden (bzgl. Performance)
  - Detaillierte Messungen für verschiedene Funktionen
  - Abhängigkeiten zwischen der SW und dem System sind identifizierbar
  - Versteckte Flaschenhälse können gefunden werden
  - Frühe Optimierungsergebnisse

1

Einleitung

2

Dynamic Performance Stubs (Übersicht)

3

Dynamic Performance Stubs (Framework)

4

Methodik CPU Stubs

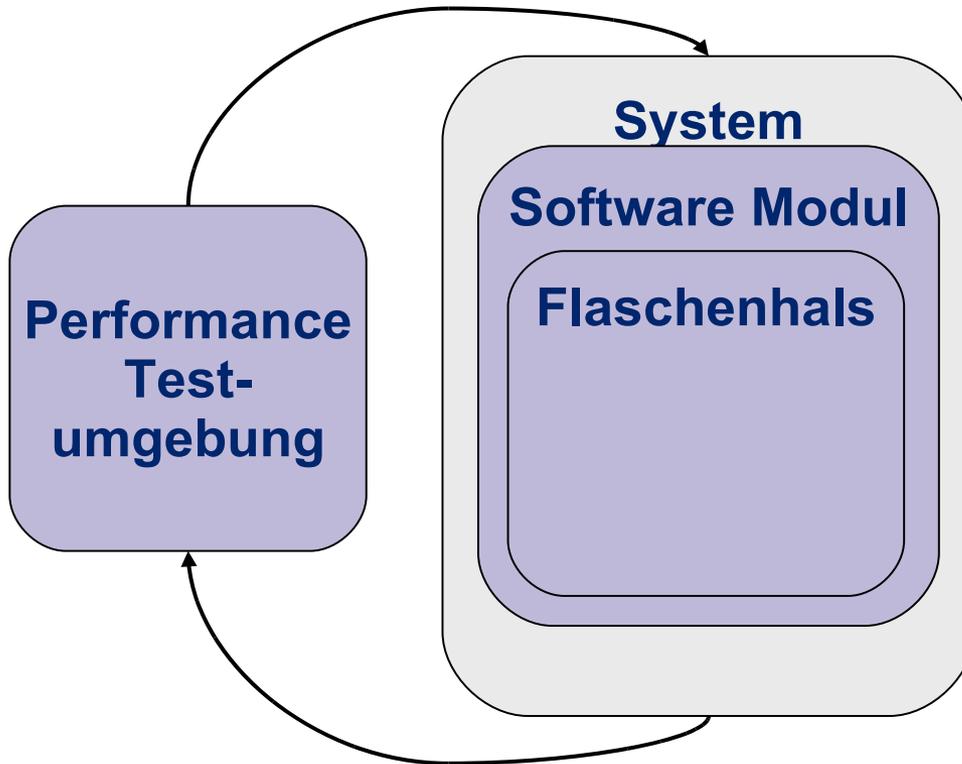
5

Fallstudie CPU Stubs

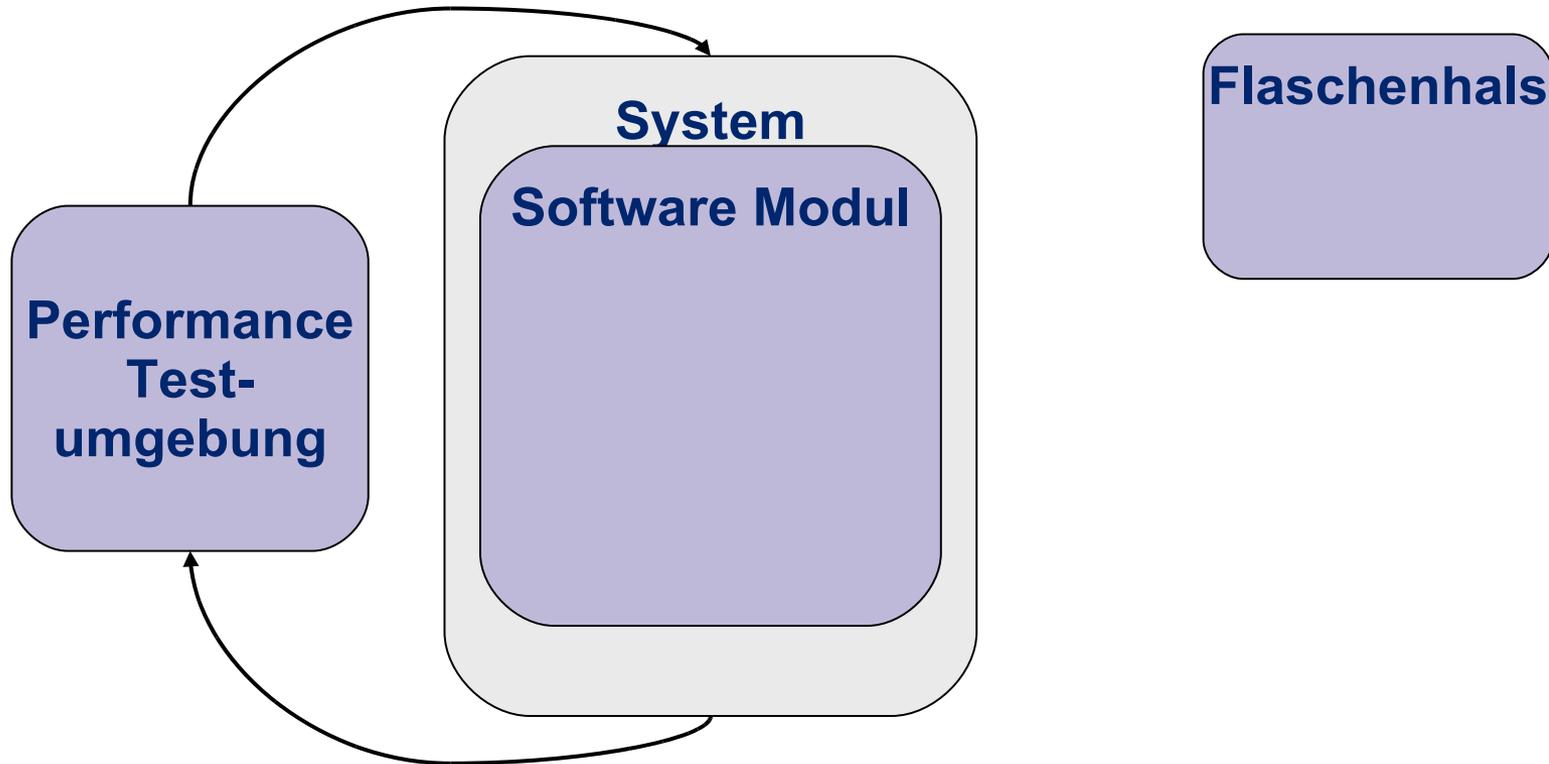
6

Zusammenfassung & Weiteres Vorgehen

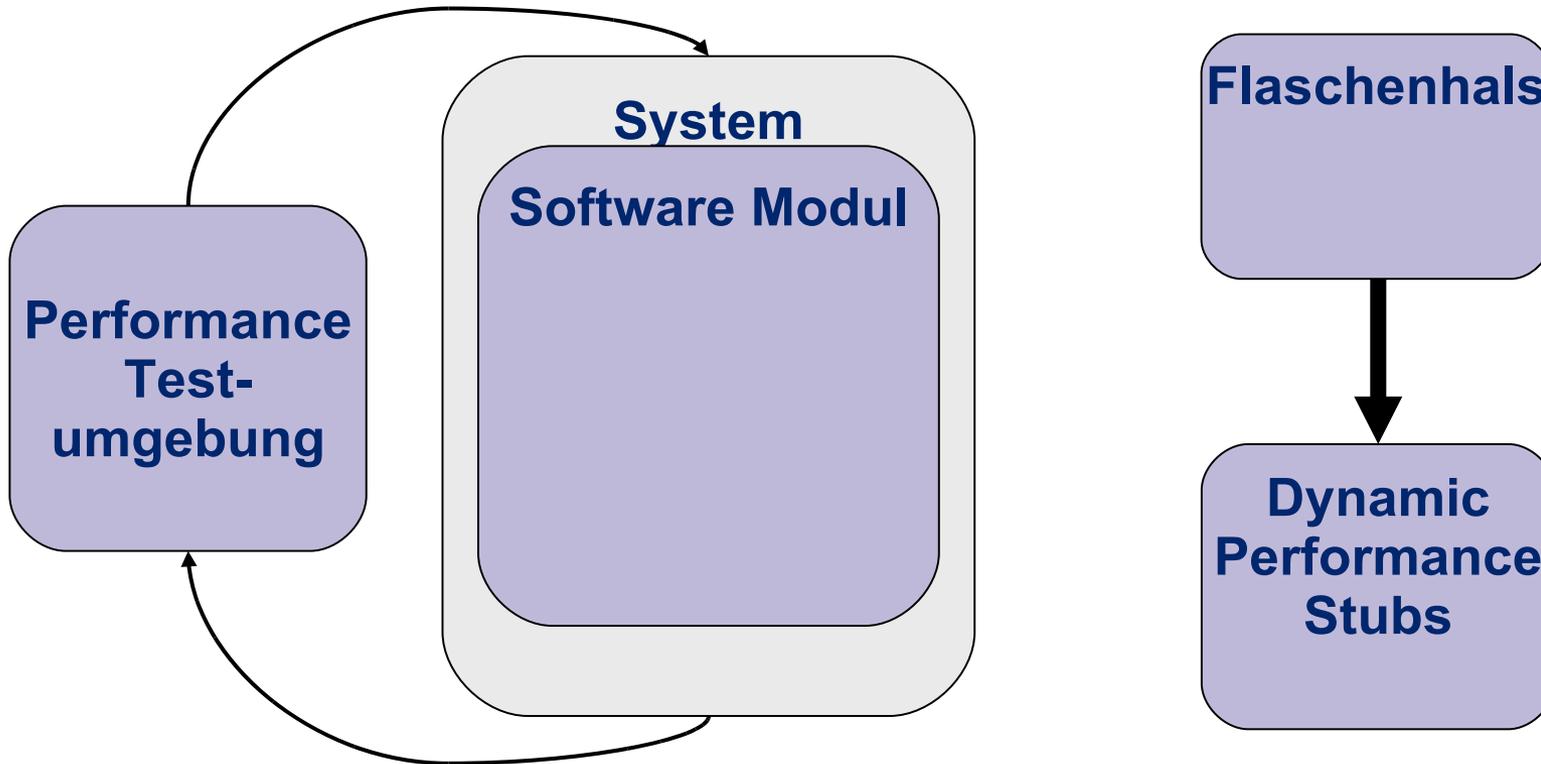
## Wie werden Dynamic Performance Stubs verwendet?



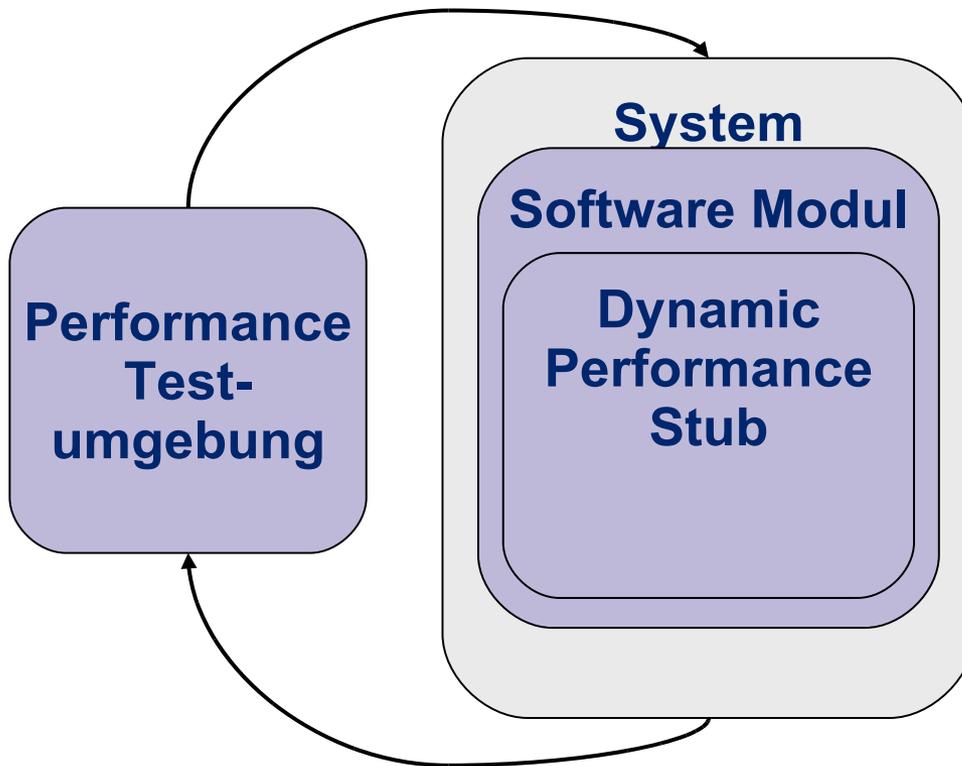
## Wie werden Dynamic Performance Stubs verwendet?



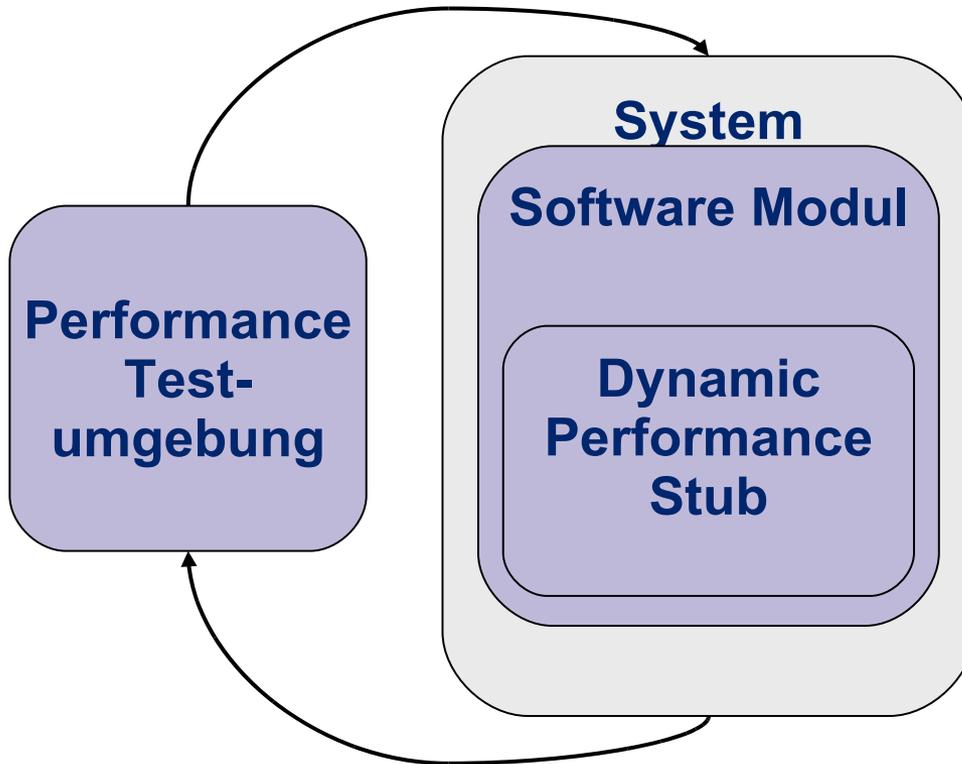
## Wie werden Dynamic Performance Stubs verwendet?



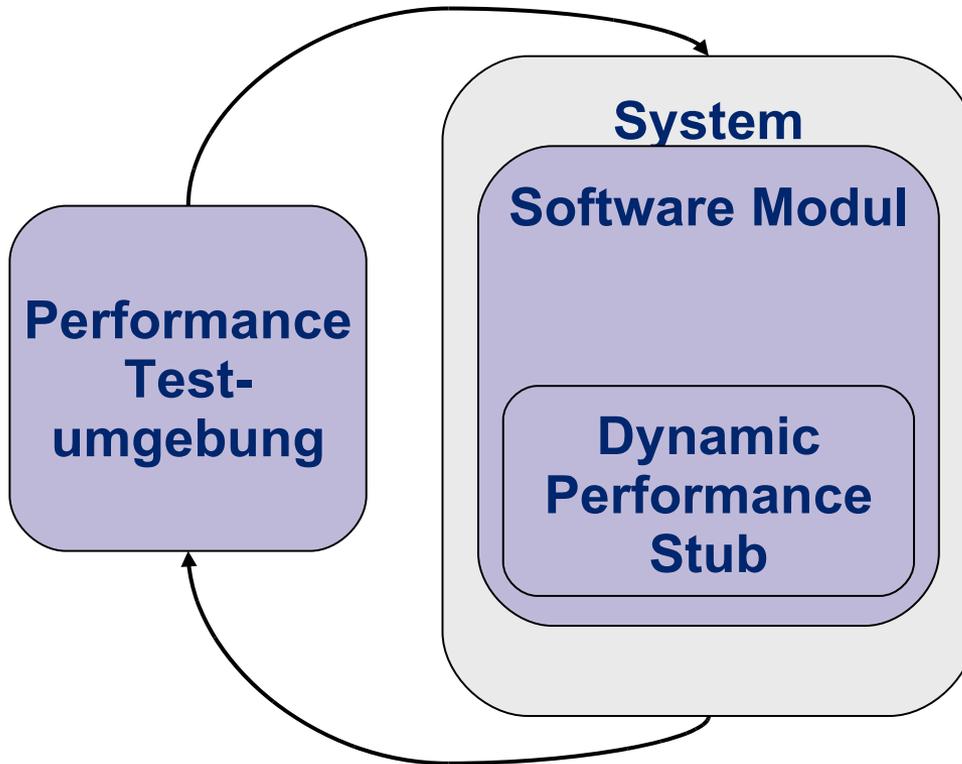
## Wie werden Dynamic Performance Stubs verwendet?



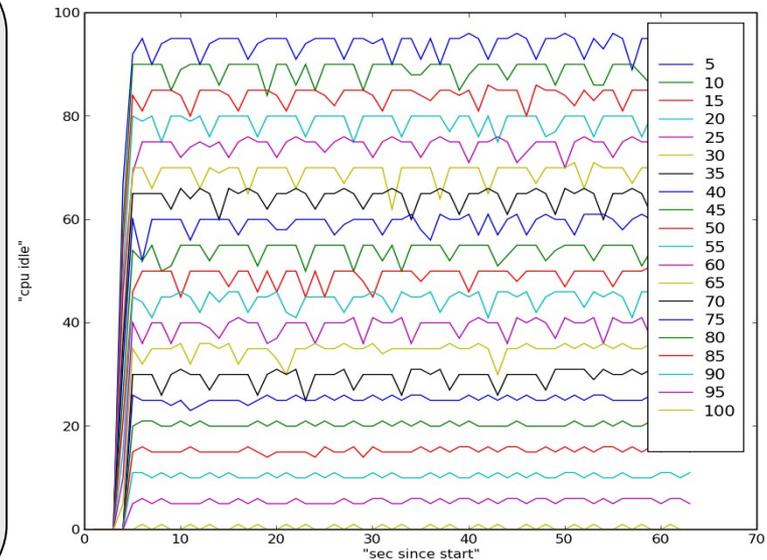
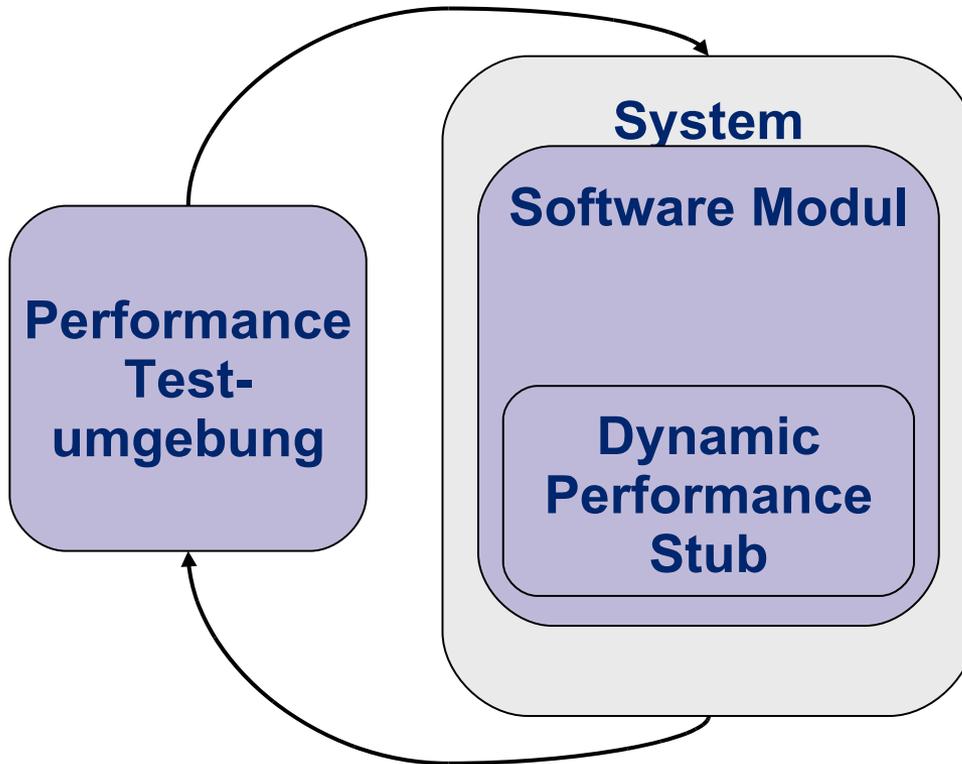
## Wie werden Dynamic Performance Stubs verwendet?



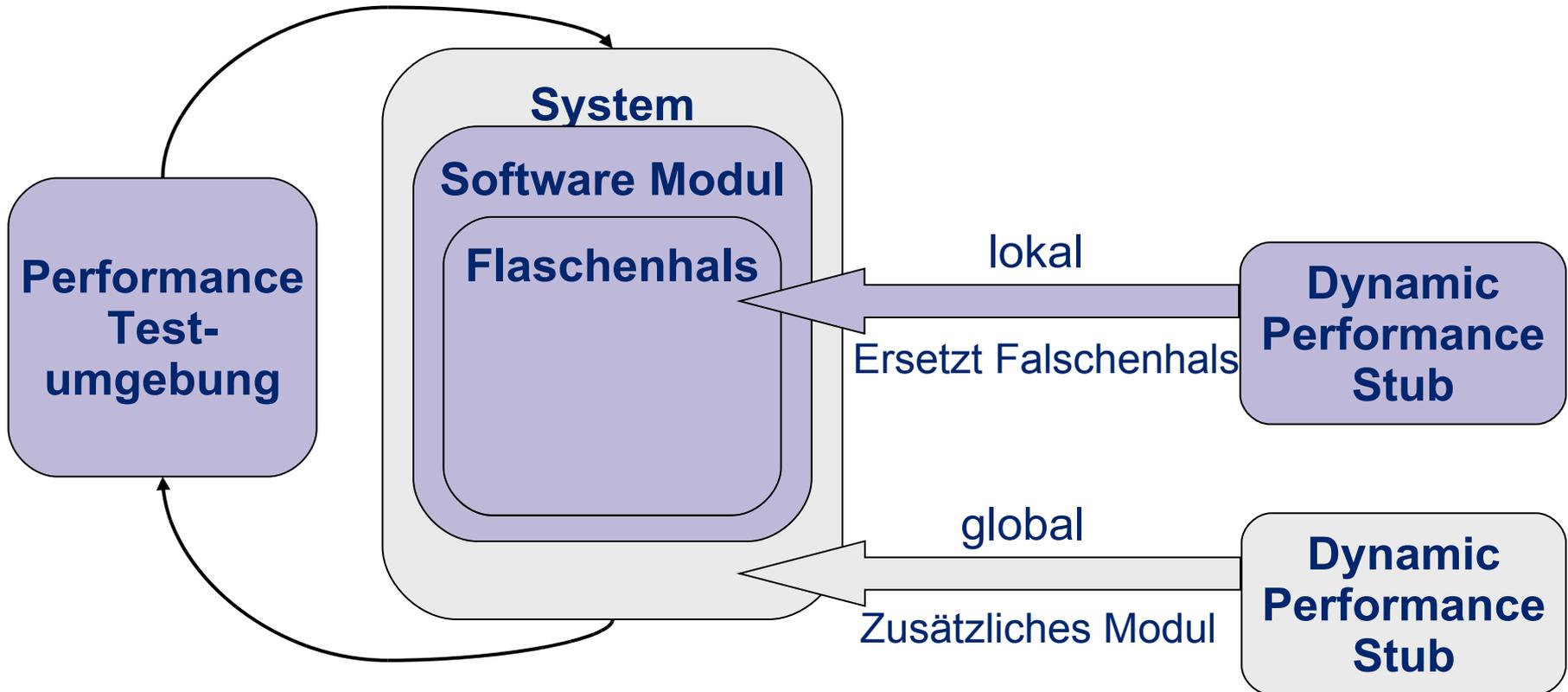
## Wie werden Dynamic Performance Stubs verwendet?



## Wie werden Dynamic Performance Stubs verwendet?



### Welche Ausprägungen gibt es?



1

**Einleitung**

2

**Dynamic Performance Stubs (Übersicht)**

3

**Dynamic Performance Stubs (Framework)**

4

**Methodik CPU Stubs**

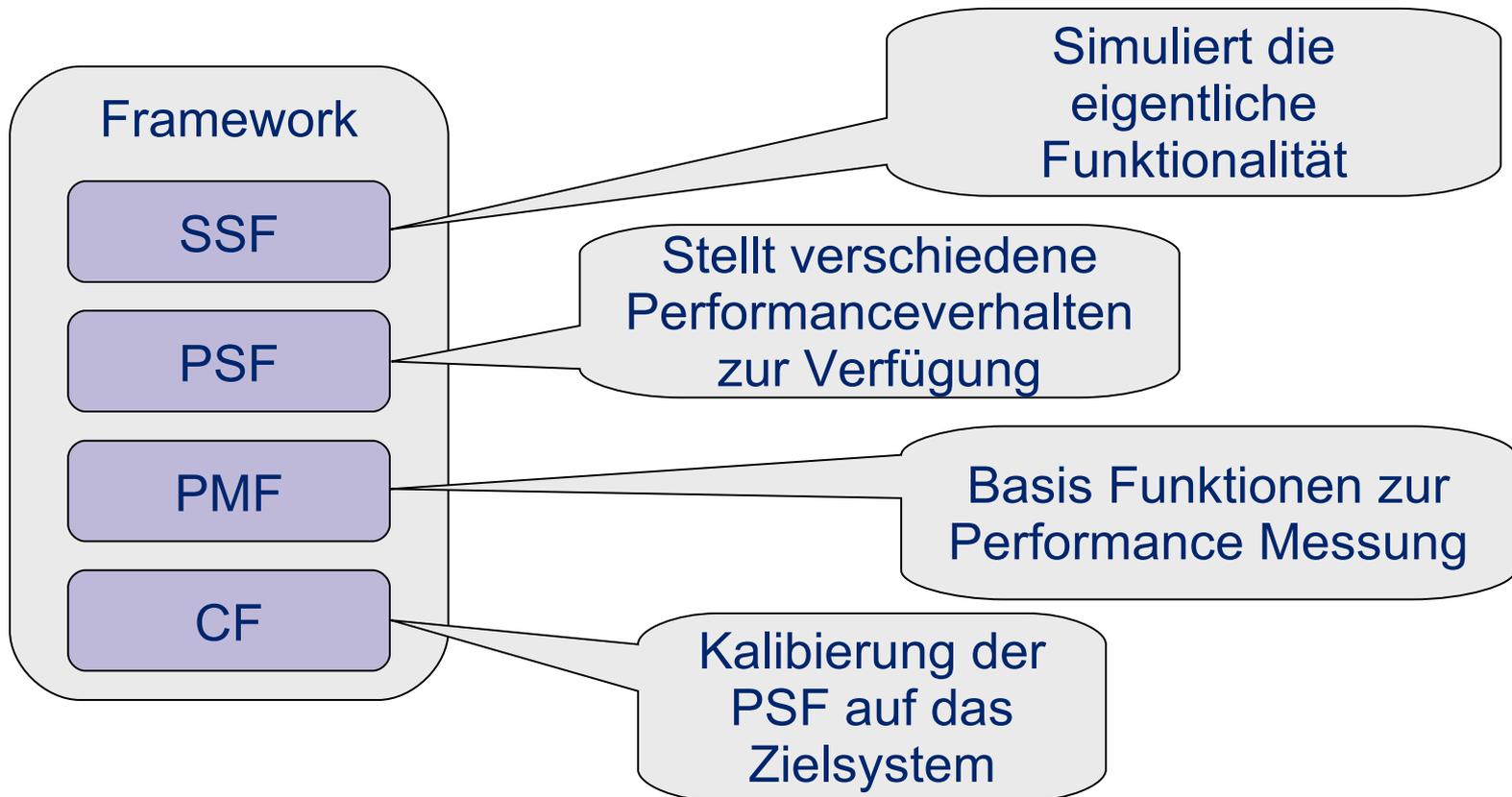
5

**Fallstudie CPU Stubs**

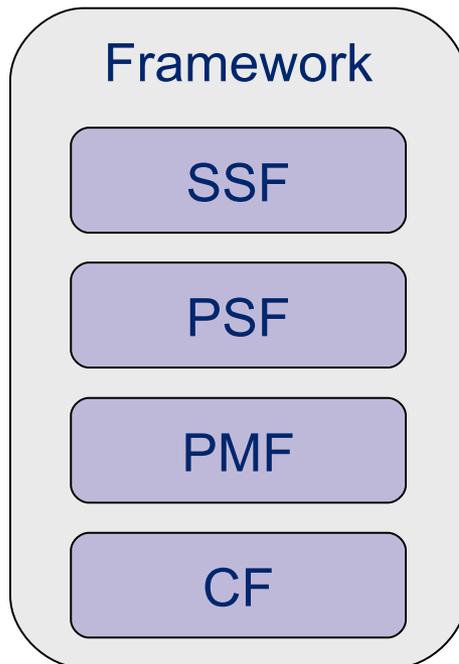
6

**Zusammenfassung & Weiteres Vorgehen**

## Was sind Dynamic Performance Stubs?



## Was sind Dynamic Performance Stubs?



Eher eine Werkzeugkiste als eine vollständig automatisierte Vorgehensweise

1

Einleitung

2

Dynamic Performance Stubs (Übersicht)

3

Dynamic Performance Stubs (Framework)

4

Methodik CPU Stubs

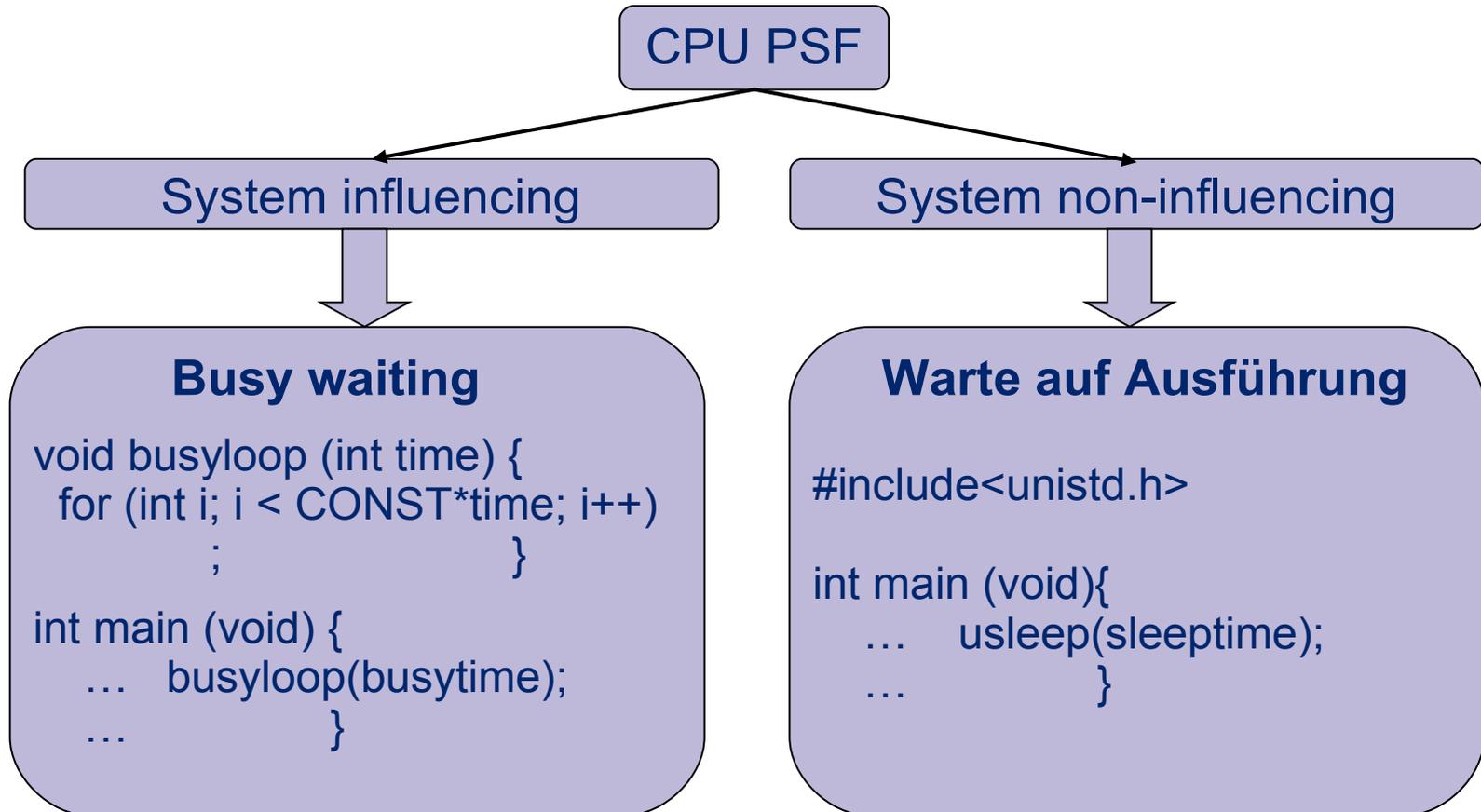
5

Fallstudie CPU Stubs

6

Zusammenfassung & Weiteres Vorgehen

## Wie werden CPU Stubs realisiert?

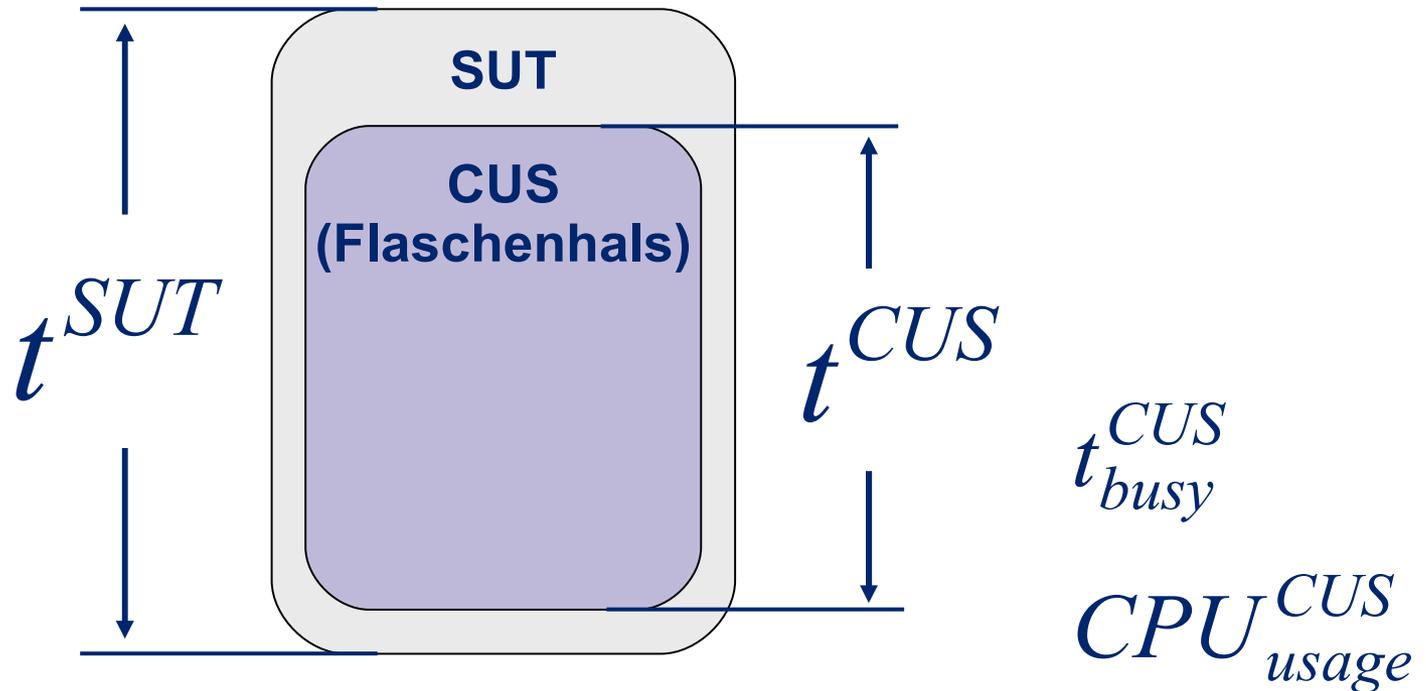


### Vorgehensweise



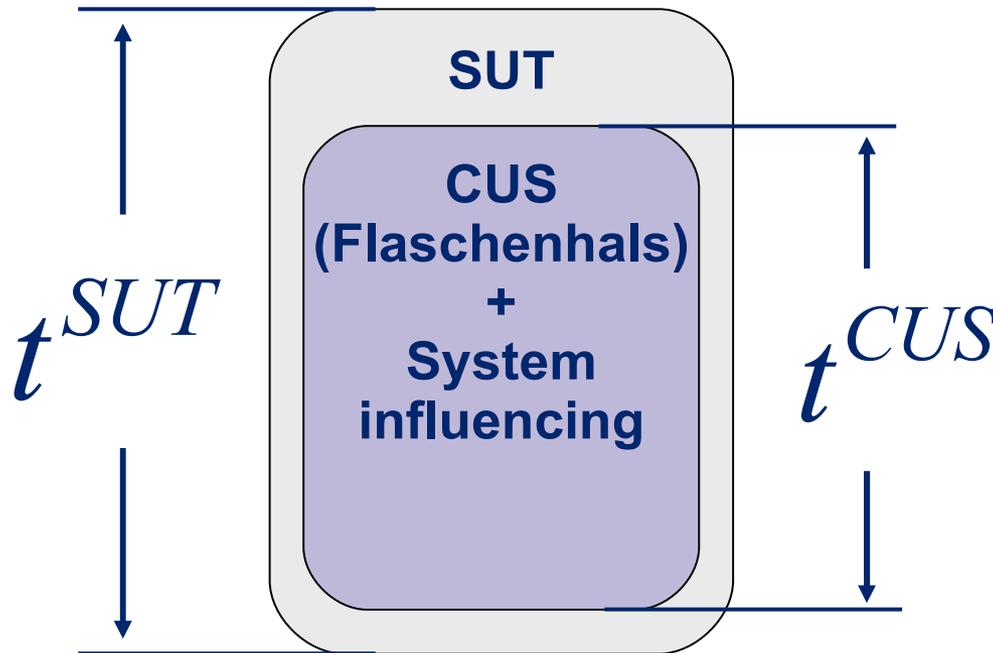
## Vorgehensweise

### 1. Flaschenhals (CUS)



## Vorgehensweise

### 2. Validierung Falschenhals



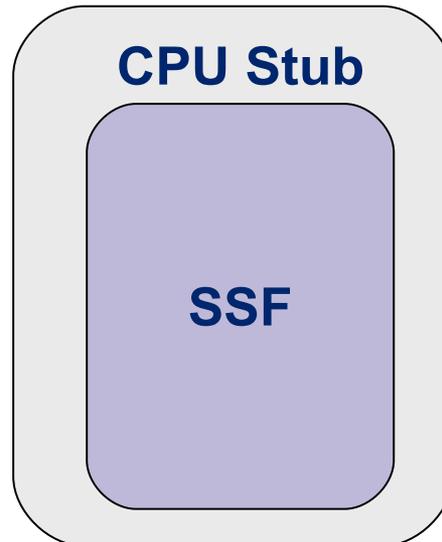
## Vorgehensweise

### 3. Untersuchung Flaschenhals

$$CPU_{usage}^{CUS} \approx 1$$

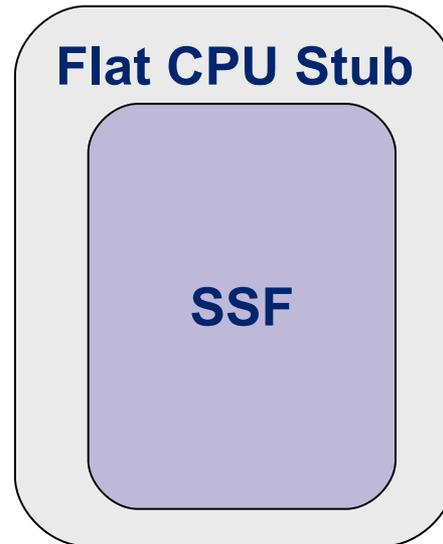
## Vorgehensweise

### 4. SSF validieren



### Vorgehensweise

#### 5. Flat CPU Stub

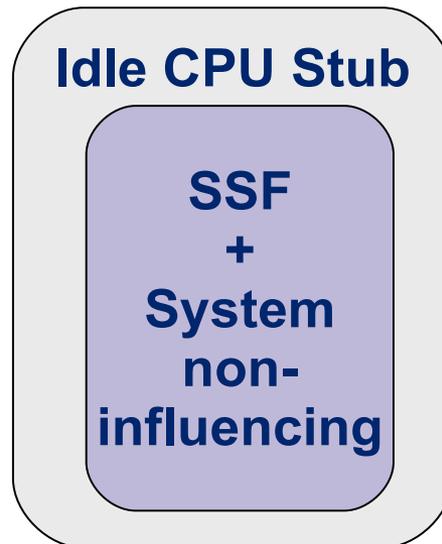


$t_{flat}^{STUB}$

$t_{perf\_gain}$

## Vorgehensweise

### 6. Idle CPU Stub



- Systemabhängigkeiten identifizieren
- Prozessabhängigkeiten identifizieren

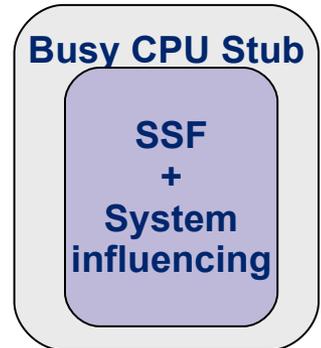
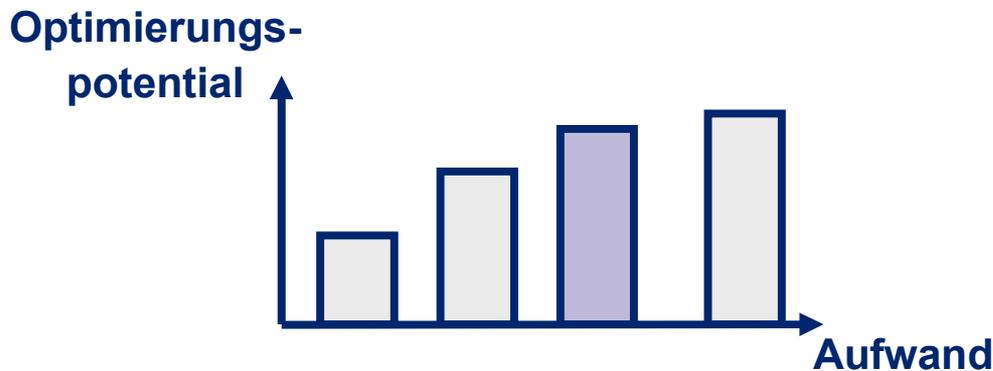
## Vorgehensweise

### 7. Optimierungsaufwand ermitteln

7.1 Optimierungspotential ermitteln

7.2 Schrittweise Änderungen, z.B. Schrittgröße 0.1

7.3 Kosten-Nutzen Analyse



1

**Introduction**

2

**Dynamic Performance Stubs (overview)**

3

**Dynamic Performance Stubs (framework)**

4

**Methodik CPU Stubs**

5

**Fallstudie CPU Stubs**

6

**Zusammenfassung & Weiteres Vorgehen**

### Hardware

- Intel® Xeon 3 GHz  
HT deaktiviert



### Software

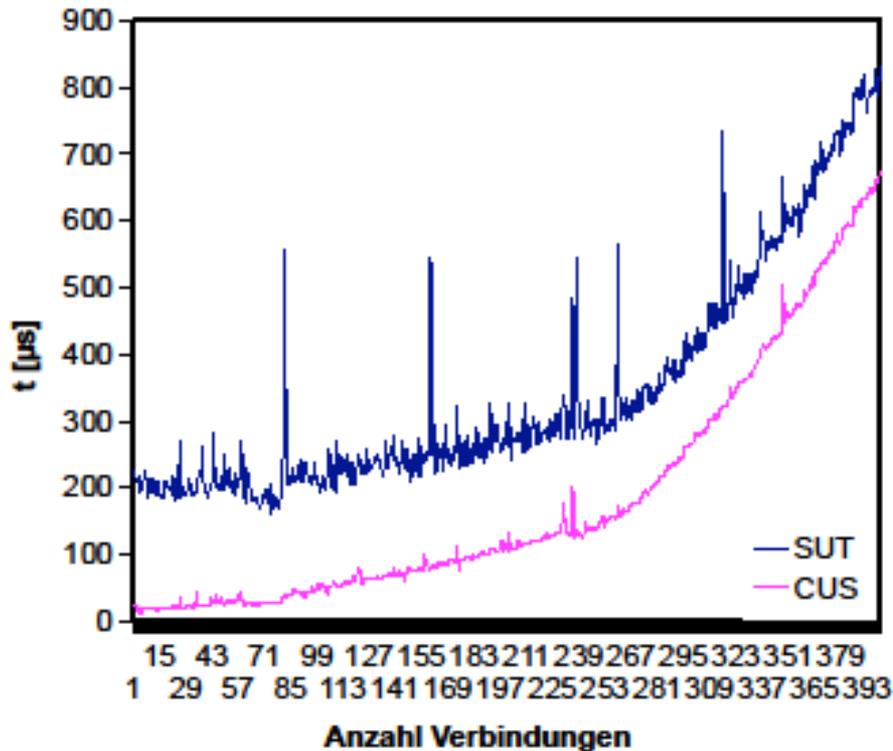
- Red Hat Enterprise Linux
- Kernel 2.6.9.22



### Evaluierung & Messungen

- Simulierte Ausführung
- Automatisierte und Wiederholbare Ausführung
- Messungen mittels „Time Stamp Counters“

### Messung des Flaschenhalses



$t^{SUT}$

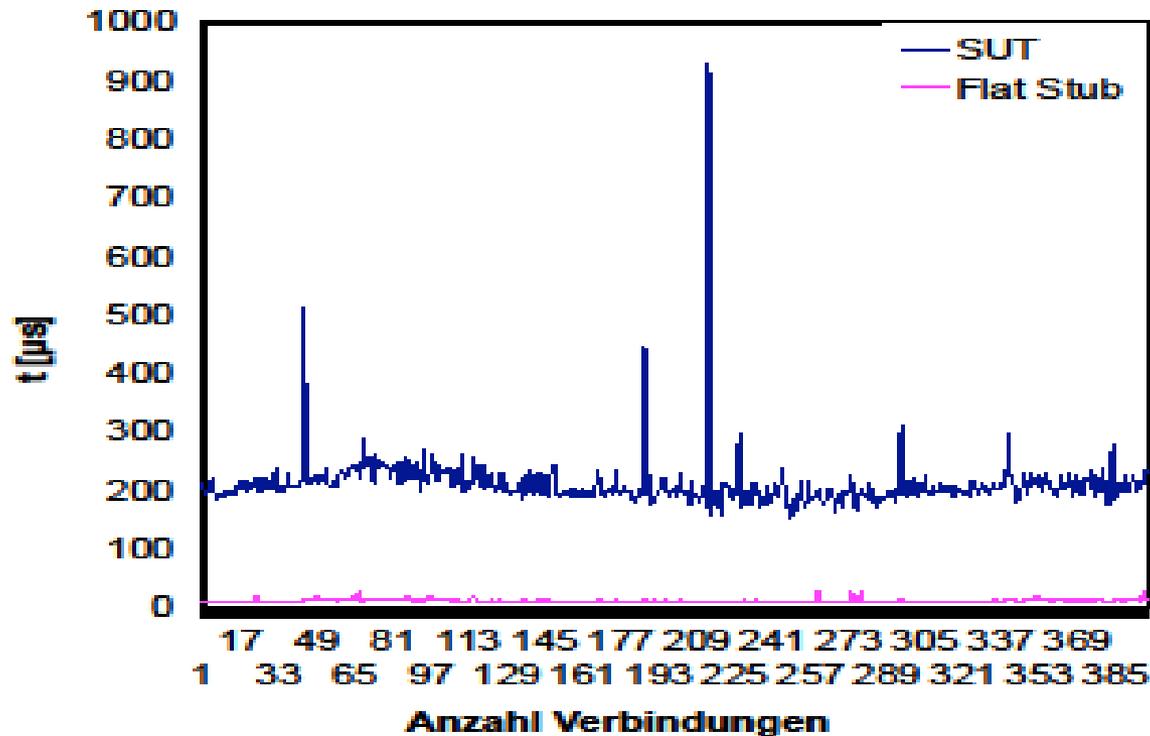
$t^{CUS}$

$t_{busy}^{CUS}$

$CPU^{CUS}$   
usage

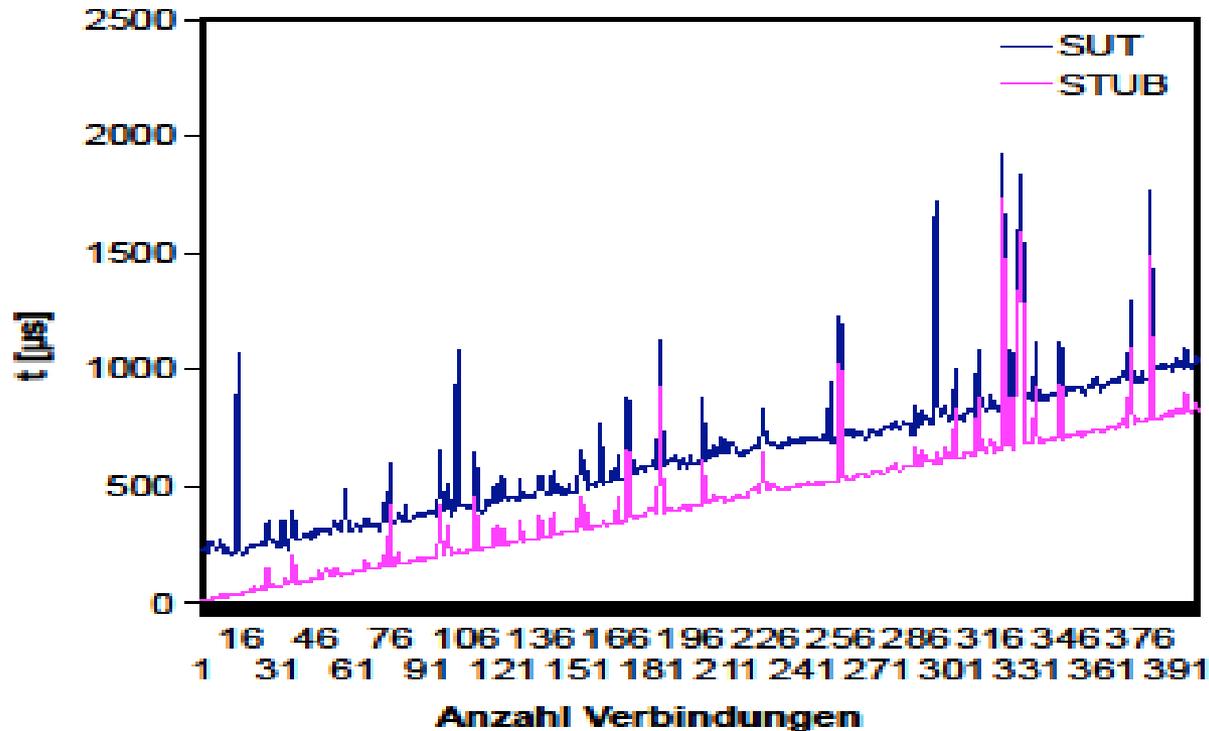
→ Schritt 1

### Messung mittels eines CPU Stubs



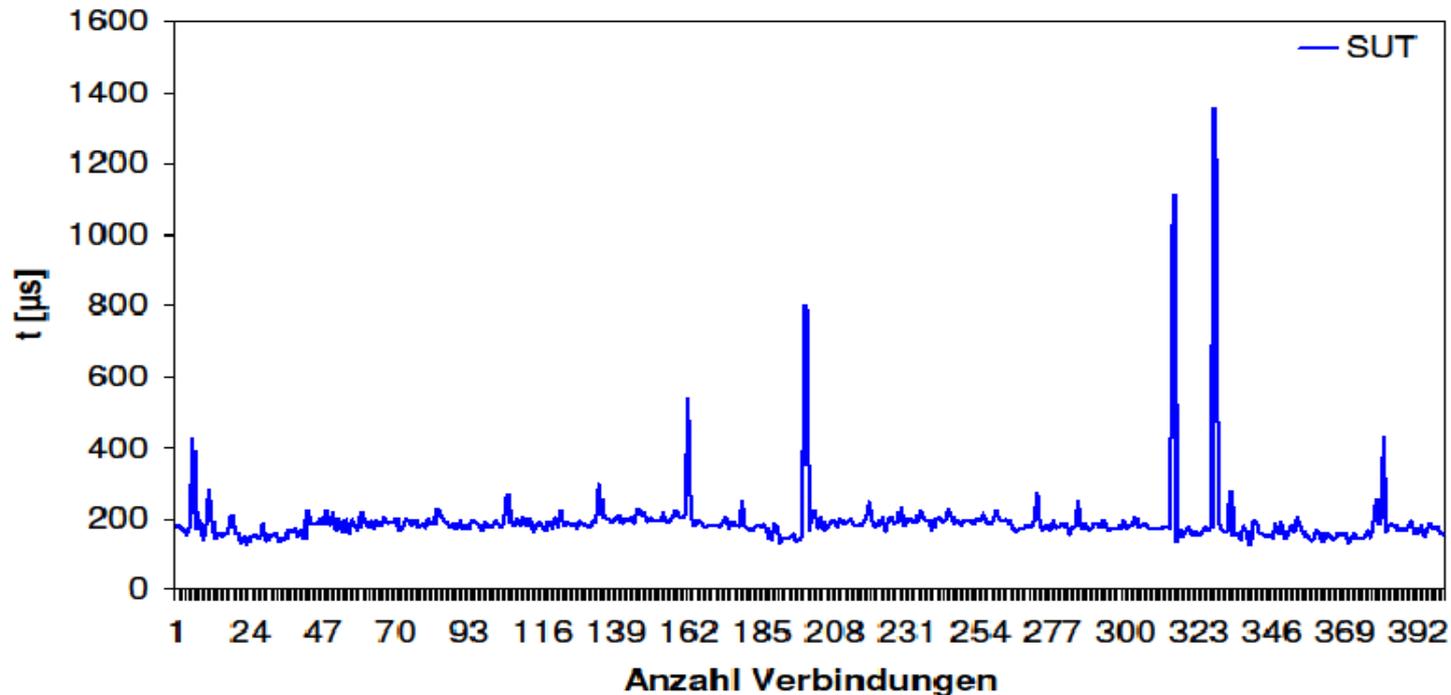
→ Schritt 5

### Messung mittels eines CPU Stubs



→ Schritt 7

### Messung nach der Optimierung



1

**Einleitung**

2

**Dynamic Performance Stubs (Übersicht)**

3

**Dynamic Performance Stubs (Framework)**

4

**Methodik CPU Stubs**

5

**Fallstudie CPU Stubs**

6

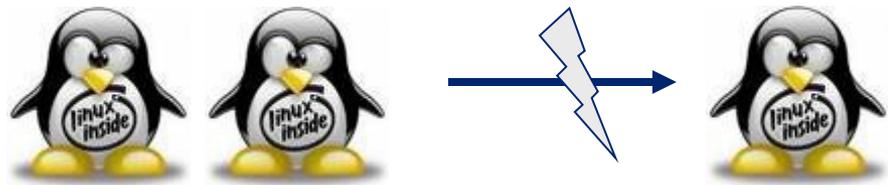
**Zusammenfassung & Weiteres Vorgehen**

## Resümee

- **Aufbau von CPU Stubs**
- **Methodik zur Verwendung von CPU Stubs**
- **Anwendung von CPU Stubs an einem Beispiel**
- **Ersetzung von Softwaremodulen oder –  
funktionen durch Stubs möglich (SSF)**

## Was ist noch offen?

- Erweitere Methodik für nicht “100%-ig” CPU gebundene Systeme
- Stubs zur Simulation weiterer Systemkomponenten, z.B.: Memory
- Überlagerung verschiedener Stubs



# Vielen Dank für Ihre Aufmerksamkeit!

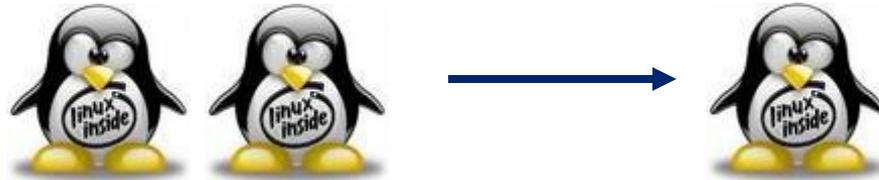
## Echtzeitverhalten durch die Verwendung von CPU Stubs: Eine Erweiterung von Dynamic Performance Stubs

Hochschule Ingolstadt  
Institut für Angewandte Forschung (IAF)

Peter Trapp  
Esplanade 10  
85049 Ingolstadt

Phone: +49-841-9348-625  
Fax: +49-841-9348-99-625  
E-mail: [peter.trapp@haw-ingolstadt.de](mailto:peter.trapp@haw-ingolstadt.de)  
www: [www.haw-ingolstadt.de/](http://www.haw-ingolstadt.de/)

Peter Trapp, 20.11.2009



# Fragen?

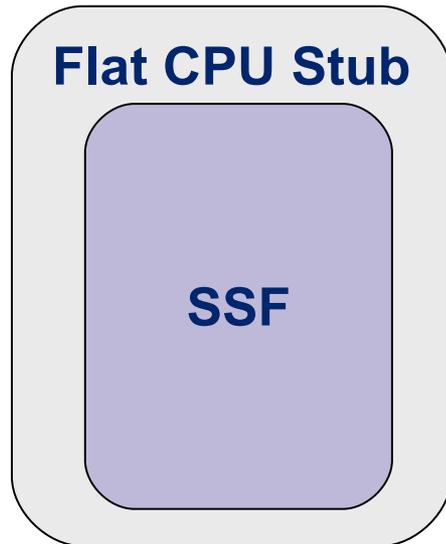
**University of Applied Sciences Ingolstadt  
Institute for Applied Research (IAF)**

Peter Trapp  
Esplanade 10  
85049 Ingolstadt

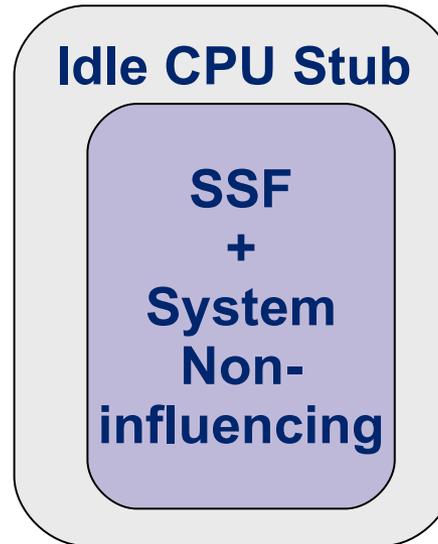
Phone: +49-841-9348-625  
Fax: +49-841-9348-99-625  
E-mail: [peter.trapp@fh-ingolstadt.de](mailto:peter.trapp@fh-ingolstadt.de)  
www: [www.fh-ingolstadt.de/iaf](http://www.fh-ingolstadt.de/iaf) Projekt: PerfOpt

## Vorgehensweise

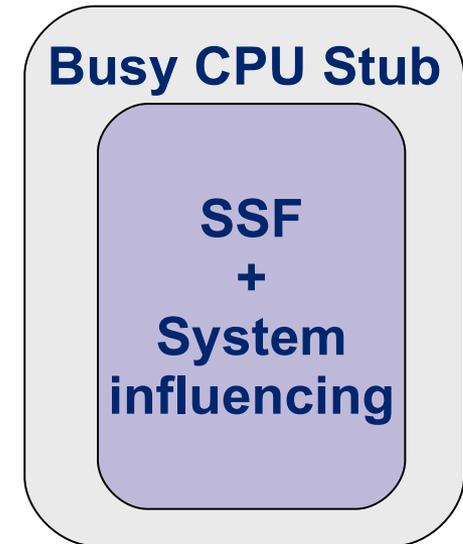
### 5. - 7. Flat / Idle / Busy CPU Stub



$t_{flat}^{STUB}$



$t_{perf\_gain}$



$t_{busy}^{STUB}$

1

**Einleitung**

2

**Dynamic Performance Stubs (Übersicht)**

3

**Dynamic Performance Stubs (Framework)**

4

**Methodik CPU Stubs**

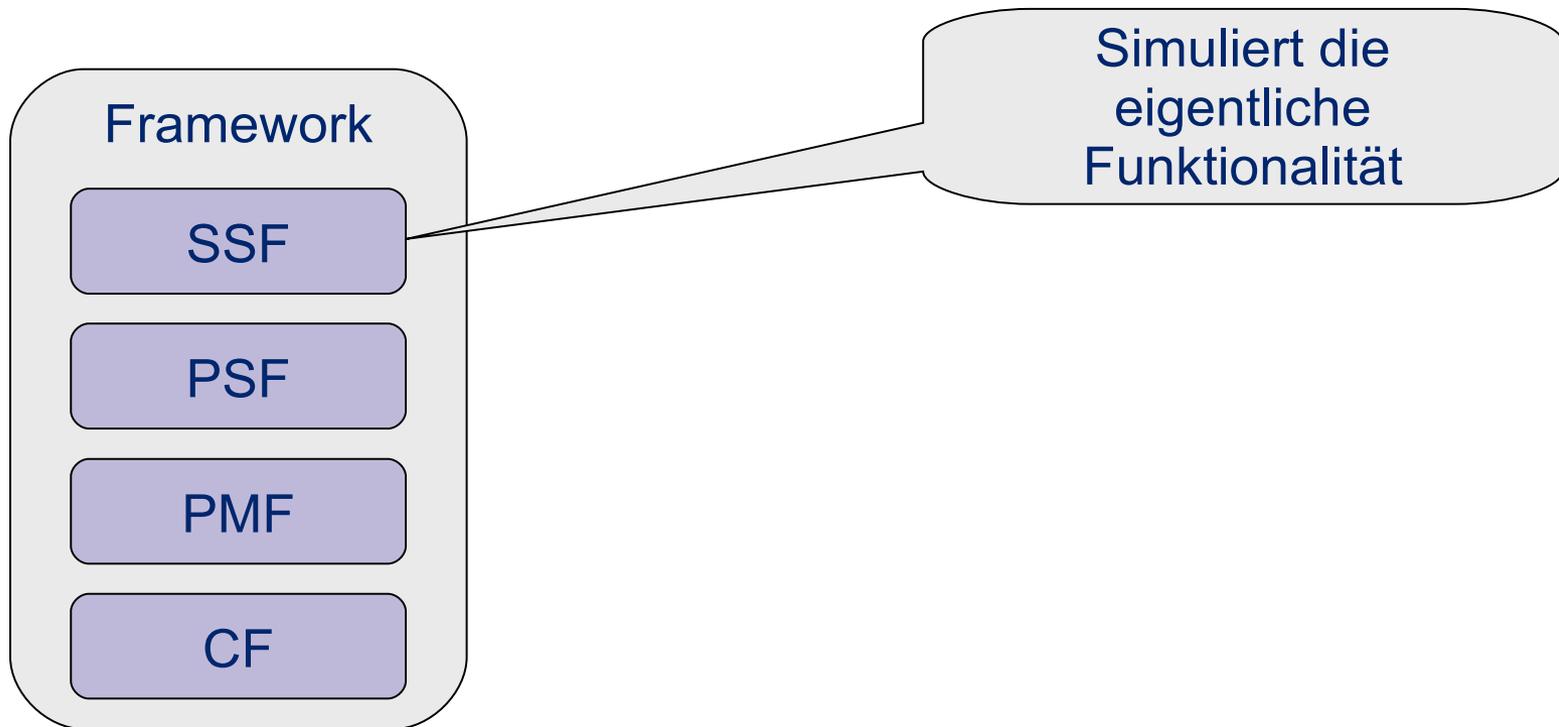
5

**Fallstudie CPU Stubs**

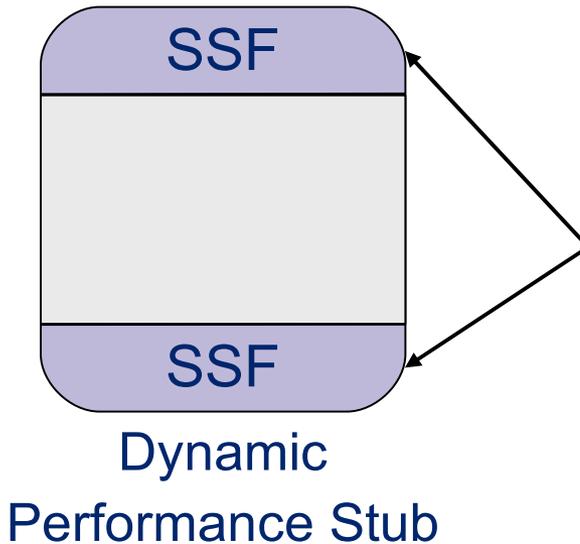
6

**Zusammenfassung & Weiteres Vorgehen**

## Was sind Dynamic Performance Stubs?

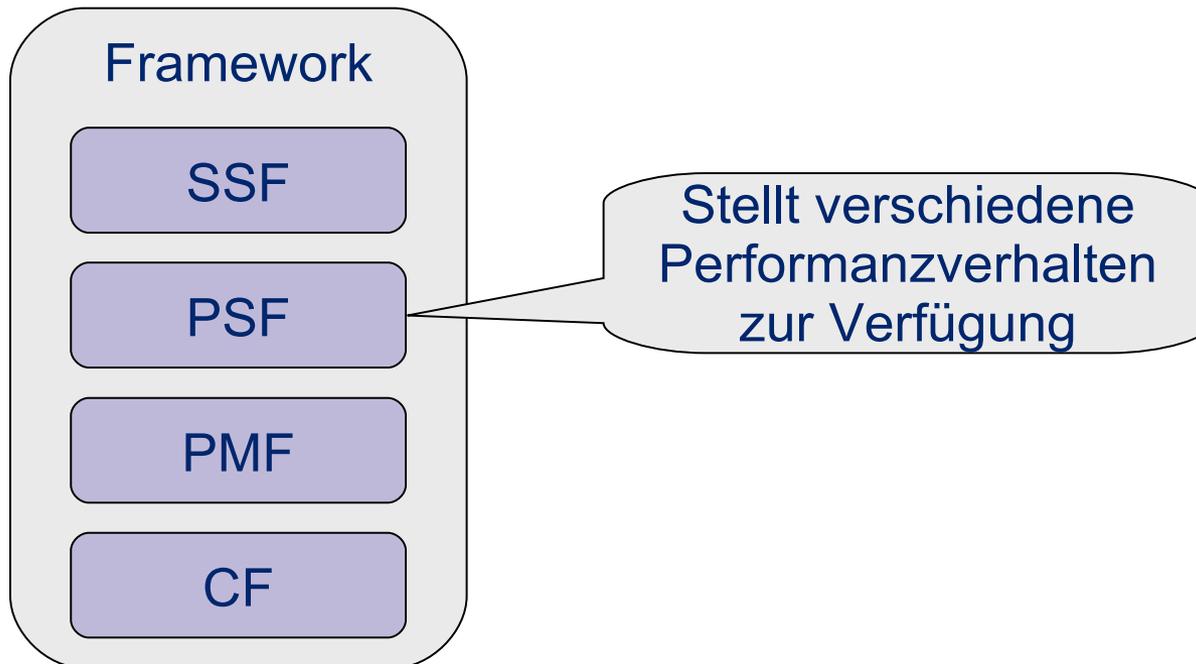


## Was ist die Simulated Software Functionality?

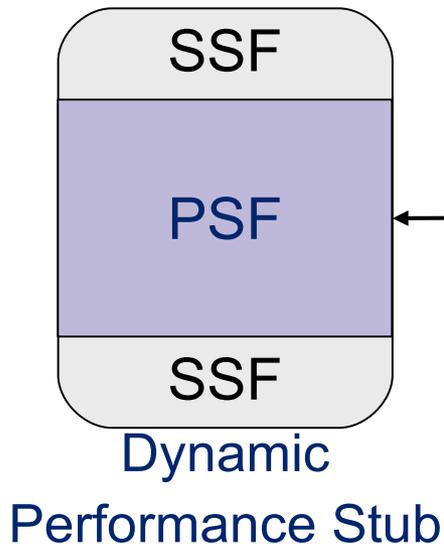


Ersetzt die Funktionalität des  
eigentlichen Moduls damit die  
Software weiterhin ausgeführt  
werden kann

## Was sind Dynamic Performance Stubs?

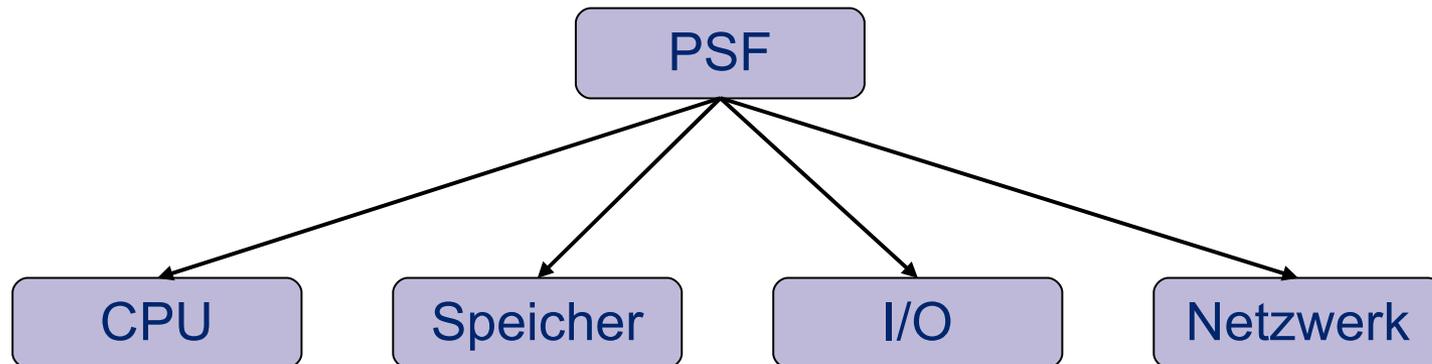


## Was sind Performance Simulation Functions?



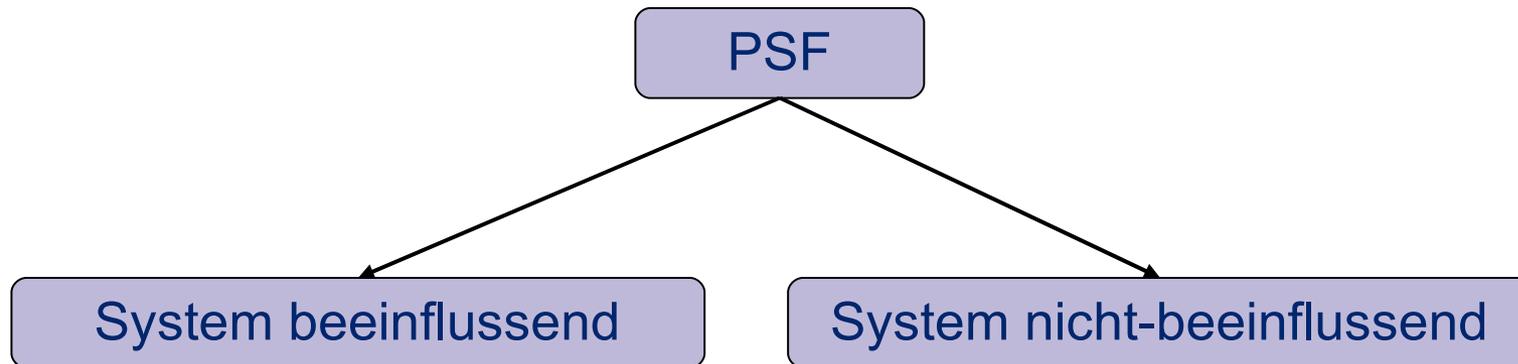
Simuliert das nicht funktionale Verhalten  
bezüglich eines Performanz Aspekts

Welches Performanzverhalten kann  
simuliert werden?

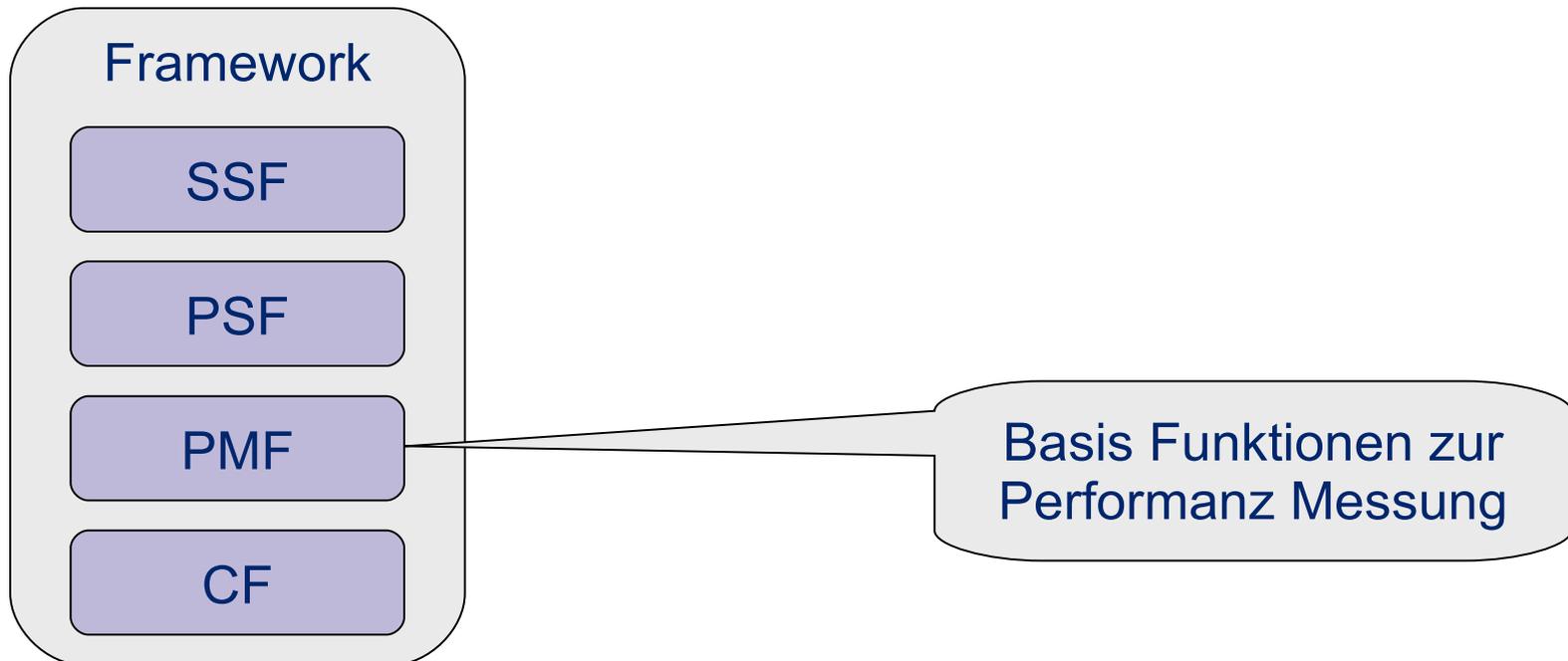




## Wie arbeiten PSF?



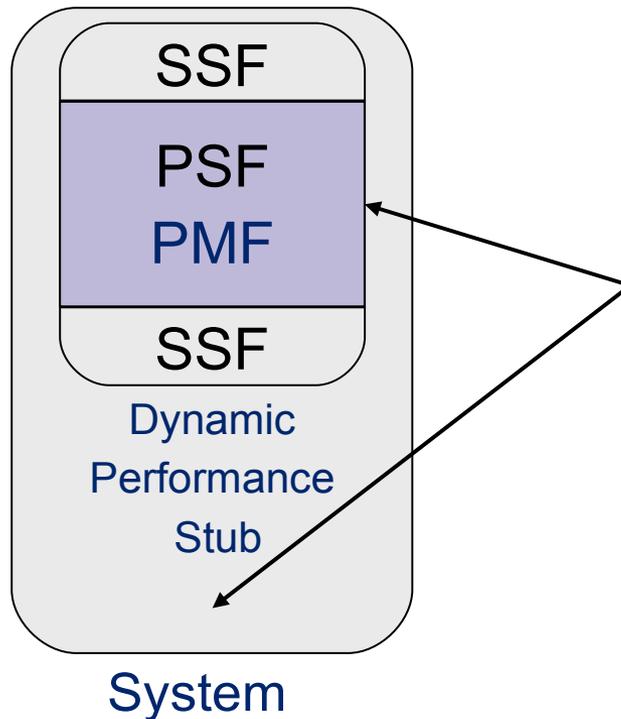
## Was sind Dynamic Performance Stubs?





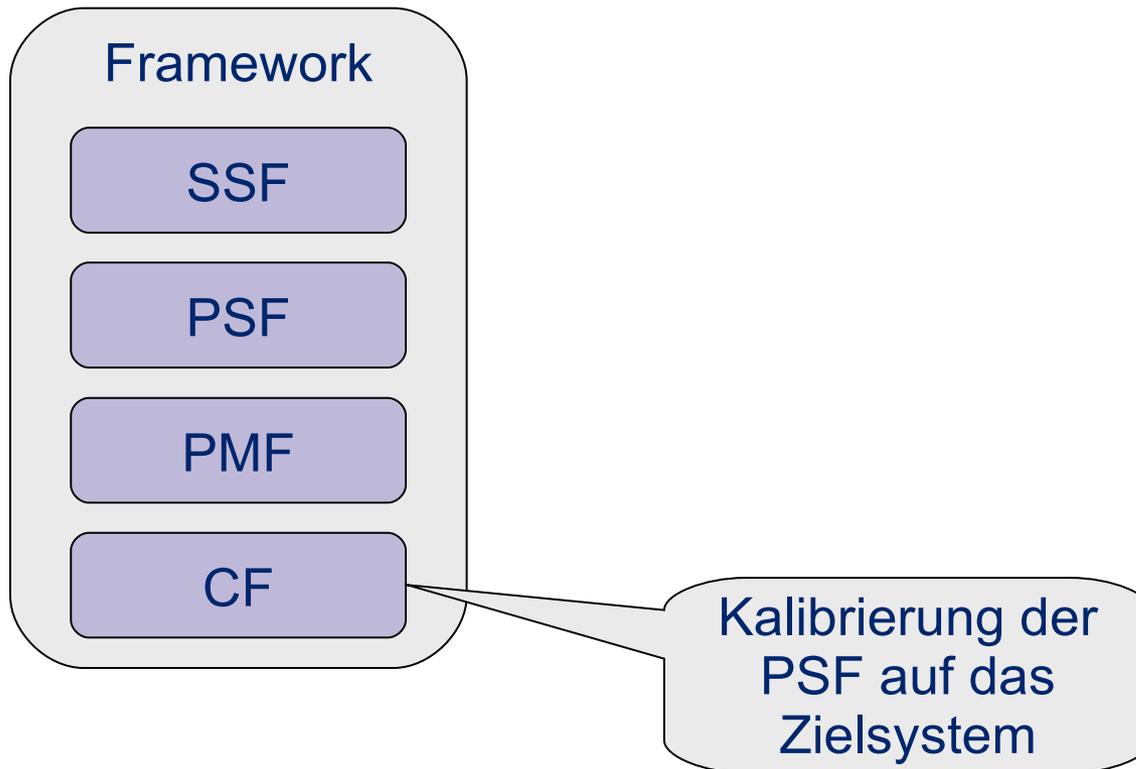
# Performance Measurement Functions I

## Was sind Performance Measurement Functions?

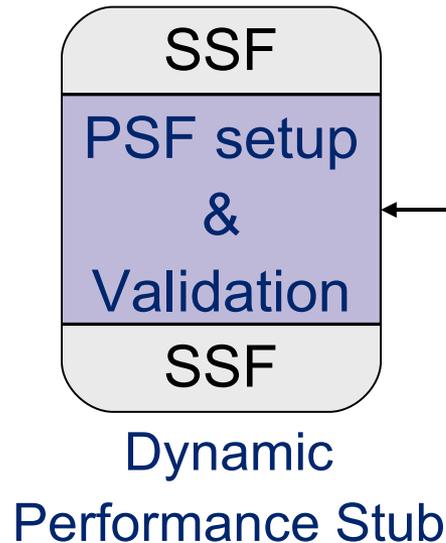


Können zur Validierung und Messung  
des Performanz Verhaltens des  
Stubs und/oder Systems verwendet  
werden.

## Was sind Dynamic Performance Stubs?



### Was sind Calibration Functions?



Führt die entsprechende PSF aus und ergänzt diese um Messungen um das Verhalten zu validieren.