

Distributed Control Lab™ at HPI

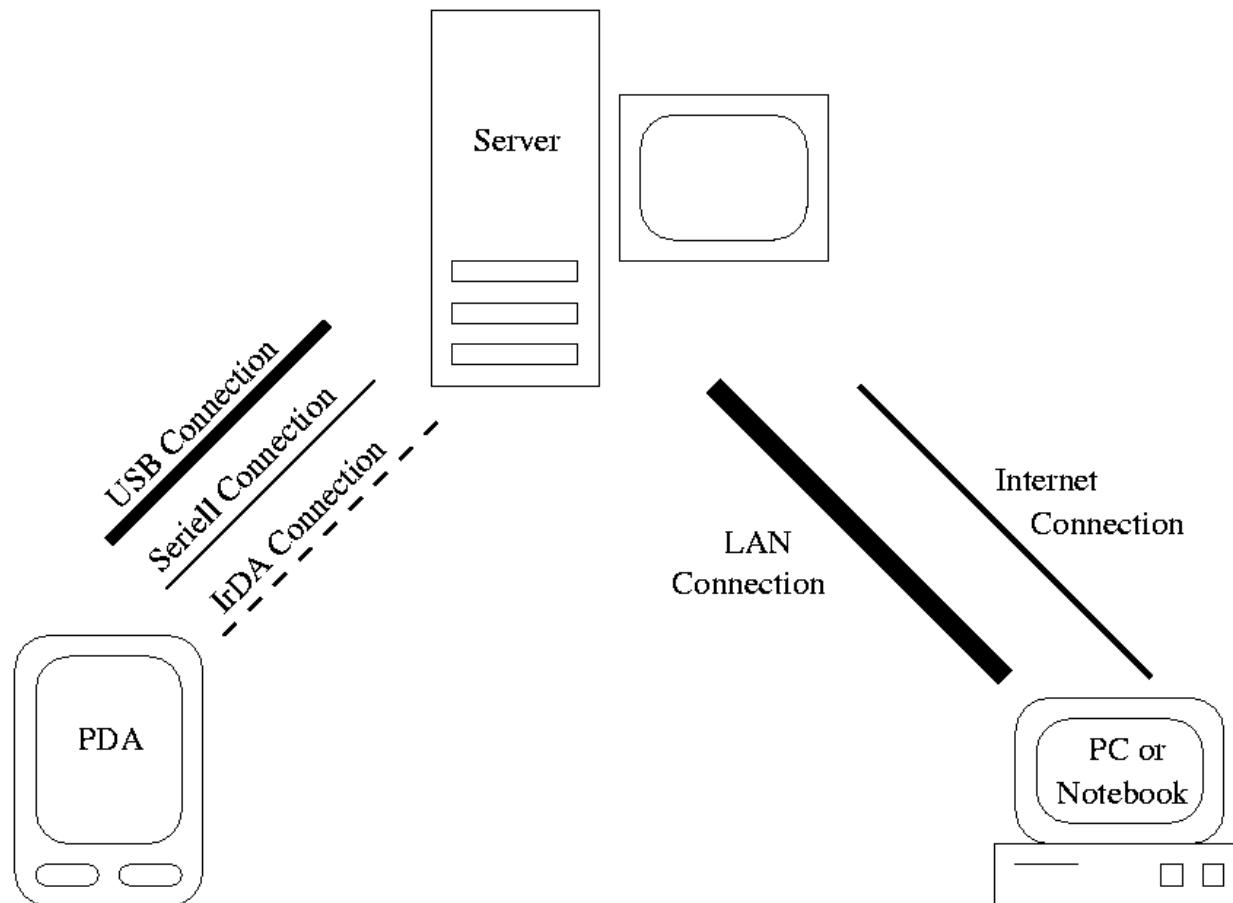
Dynamic (Re-) Configuration
as Safeguard Mechanism
in dynamically changing
environments

Overview and Motivation

Mobile systems are challenging:

- Dynamically changing parameters of communication
(Quality-of-Service)
- Varying computational power
 - among different mobile end systems
(Notebook, Pocket PC, cellular phone)
- One-size-fits-all type of services are not acceptable
- State of the art: device-specific information encoding
 - Different representations for the same device are typically not supported (i.e., image vs. text vs. voice)
 - No chance to react on changing communication settings

The Problem illustrated



Towards Configurable ad-hoc Services for Mobile Users

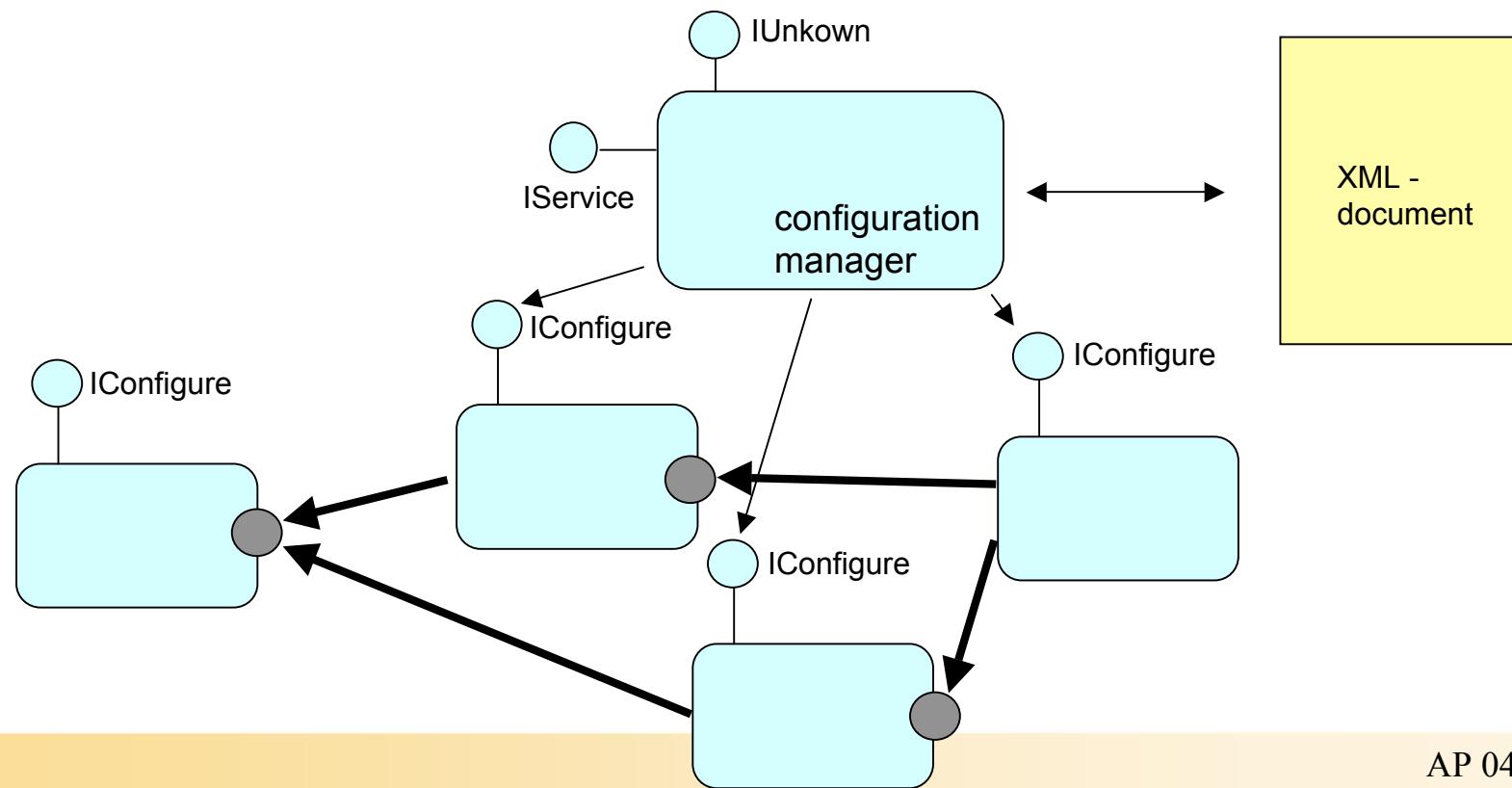
- XML-based description language for system configuration
 - Description of service profiles
- Separation of implementation and configuration
- Toolset for automatic system configuration
- Support for mobile devices in networks with dynamically changing communication settings
- Proof-of-concept scenario using COM+ and Windows CE

How to Configure a Service?

- Service: component-based software which delivers information upon client's request
- Services are made of:
 - Components + attributes (i.e.; jpeg-filter with given compression)
 - Connectors (i.e.; RPC, shared memory interconnect, TCP/IP socket)
- Service configuration consists of:
 - Component instantiation and configuration
 - Selection of the appropriate interconnection mechanism (plumbing)
- XML-based configuration description
 - Rules for dynamic re-configuration when environment changes

Configuration Process

- Configuration manager interconnects components as described in the chosen configuration profile

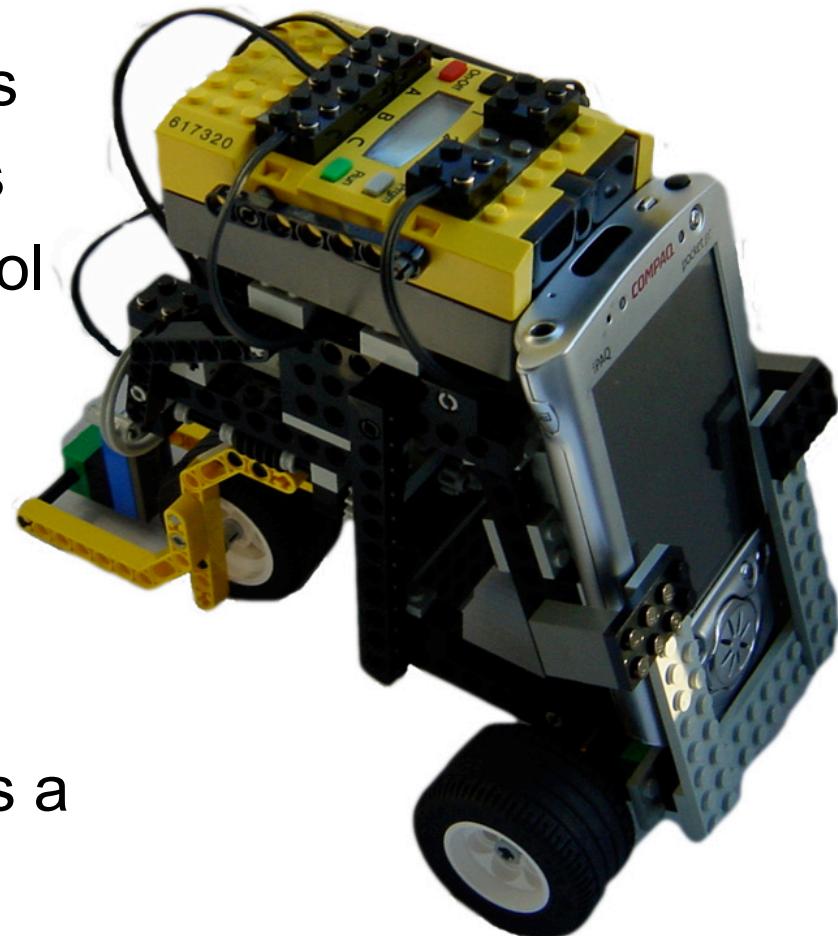


The Distributed Control Lab – a configurable robotic system

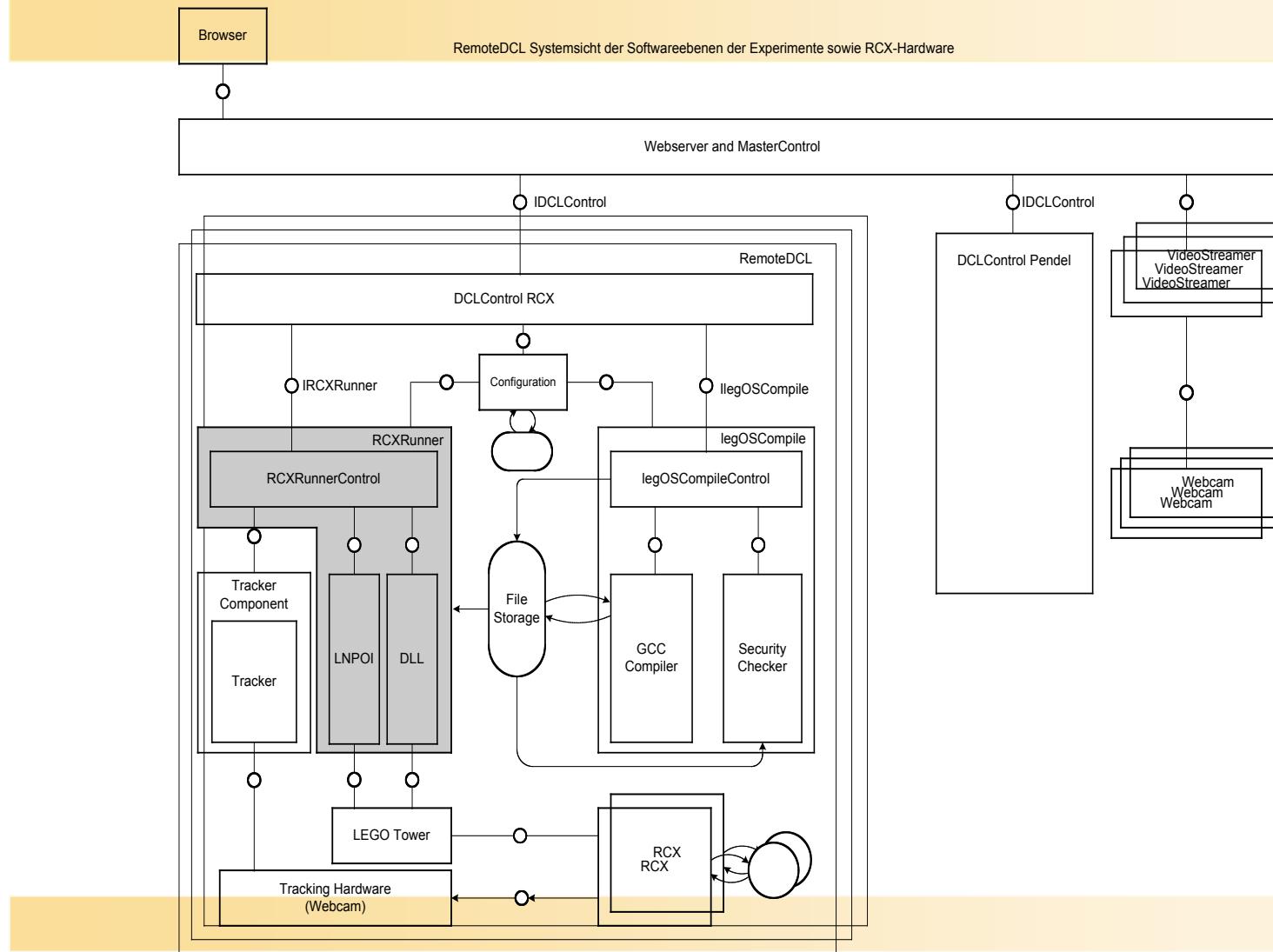
Variety of robot configurations

- Different actuators/sensors
- Experimentation with control algorithms
- Coordinated actions of multiple robots

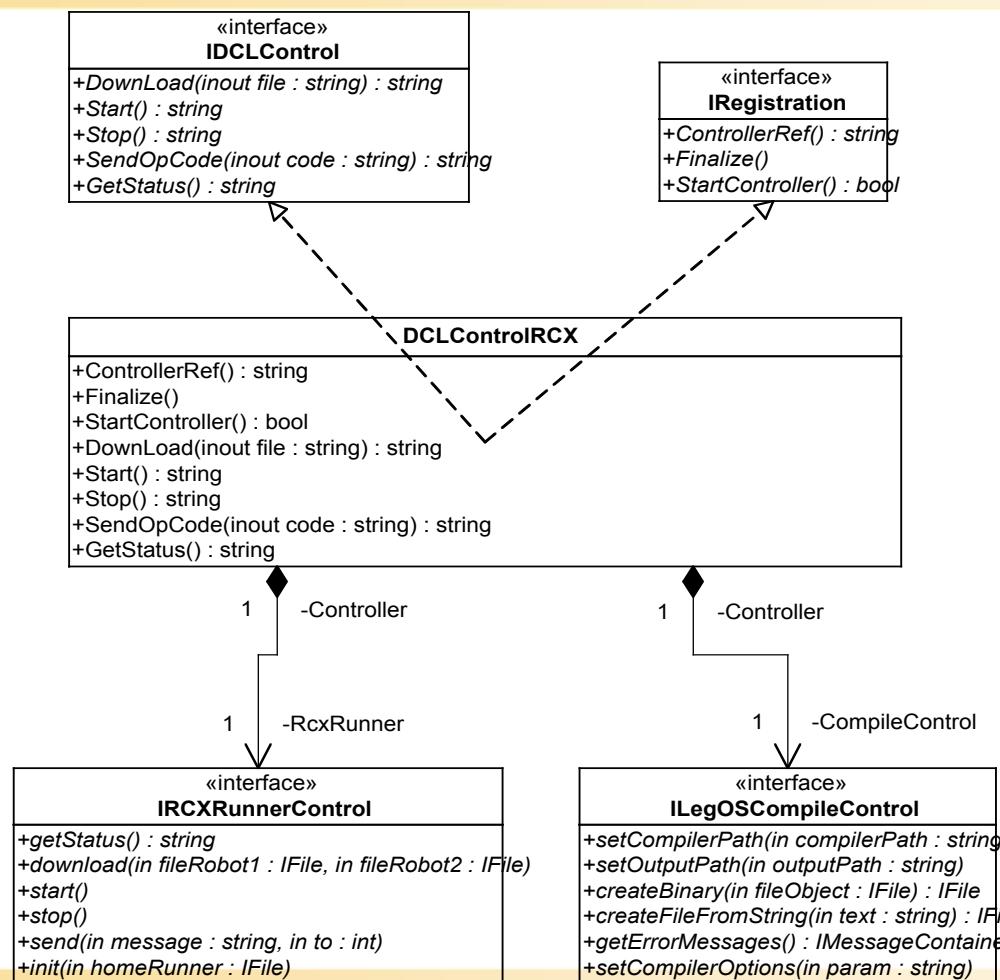
Dynamic reconfiguration as a safeguard mechanism



Distributed Control Lab – Structure



Distributed Control Lab – Interfaces



Web-based Robotics Experiments

Livebild:



Status:

Bitte geben Sie oben den Quellcode für den Roboter ein.
Wählen sie einen Programmplatz

ProgrammCode:

```
task main()
{
    OnRev(OUT_C);
    Wait(100);
    Off(OUT_C);
    OnFwd(OUT_C);
    Wait(100);
    Off(OUT_C);
}
```

Download **Start** **Stop**

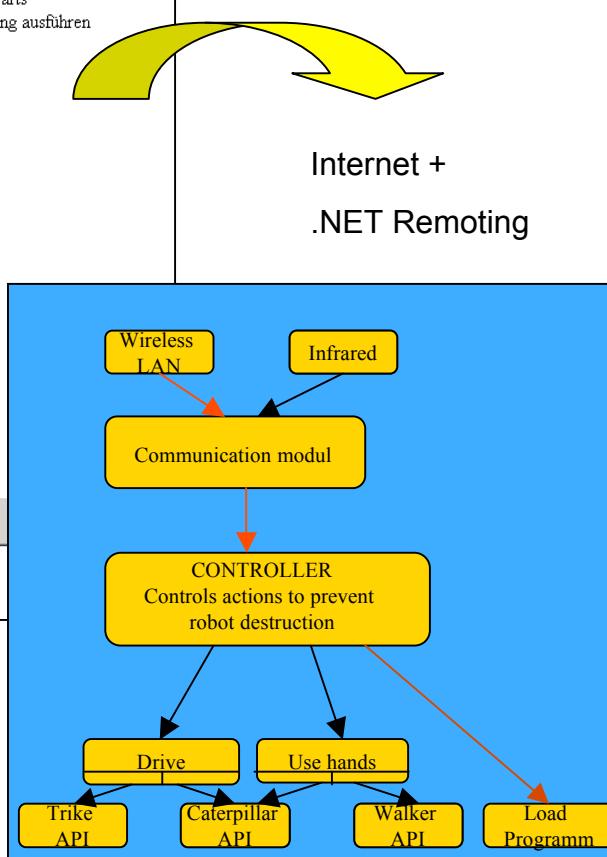
Beispielprogramme:

Vorwärts **Zurück** **Hin und Her**

Welcome to the Distributed Control Lab !!!

NQC-Befehlsreferenz

```
task main()
{
    • OnFwd(OUT_C); Vorwärts
    • OnRev(OUT_C); Rückwärts
    • Wait(Zeit); Aktion Zeit lang ausführen
}
```



ASP.NET & .NET Remoting

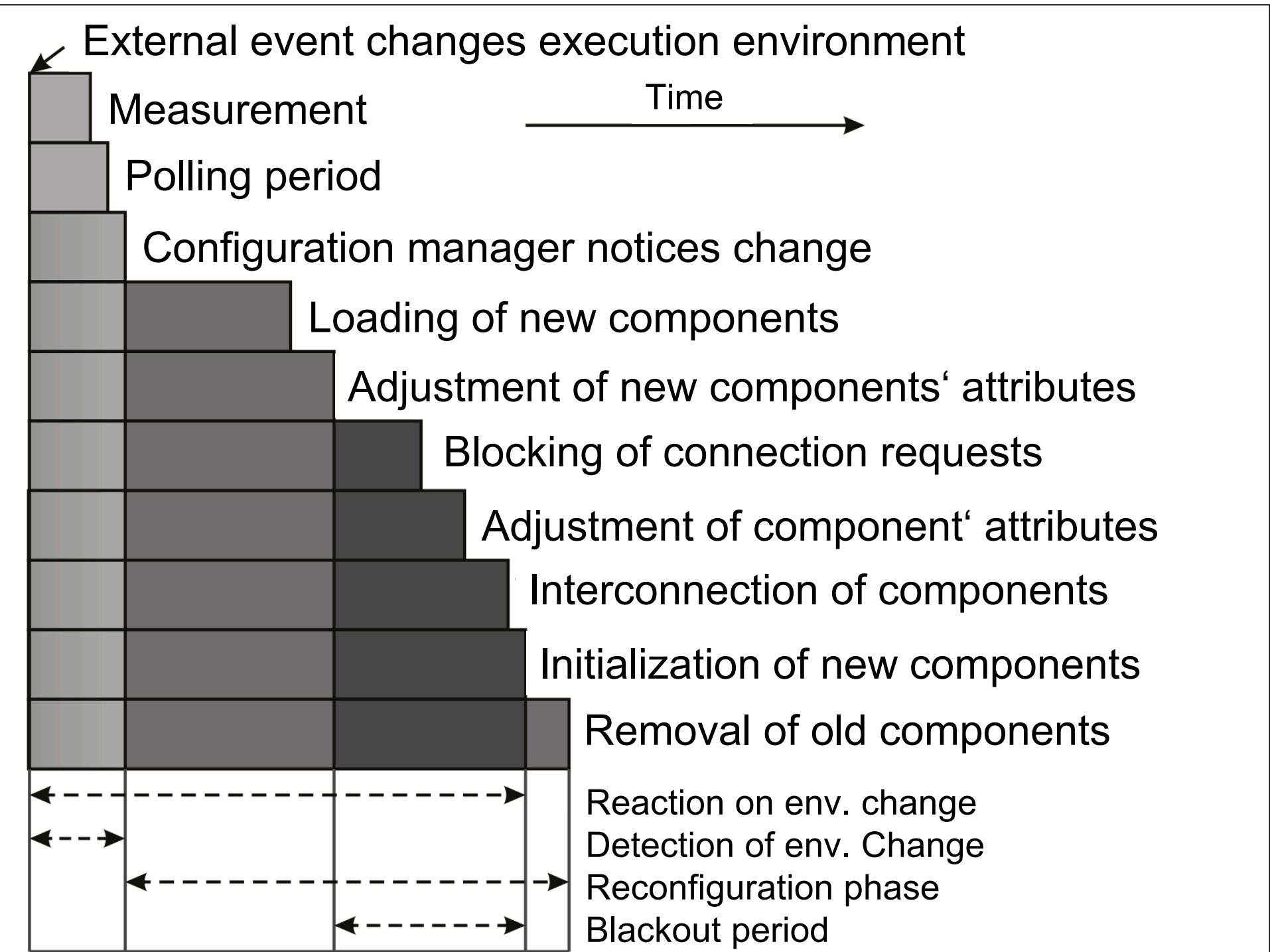
www.dcl.hpi.uni-potsdam.de

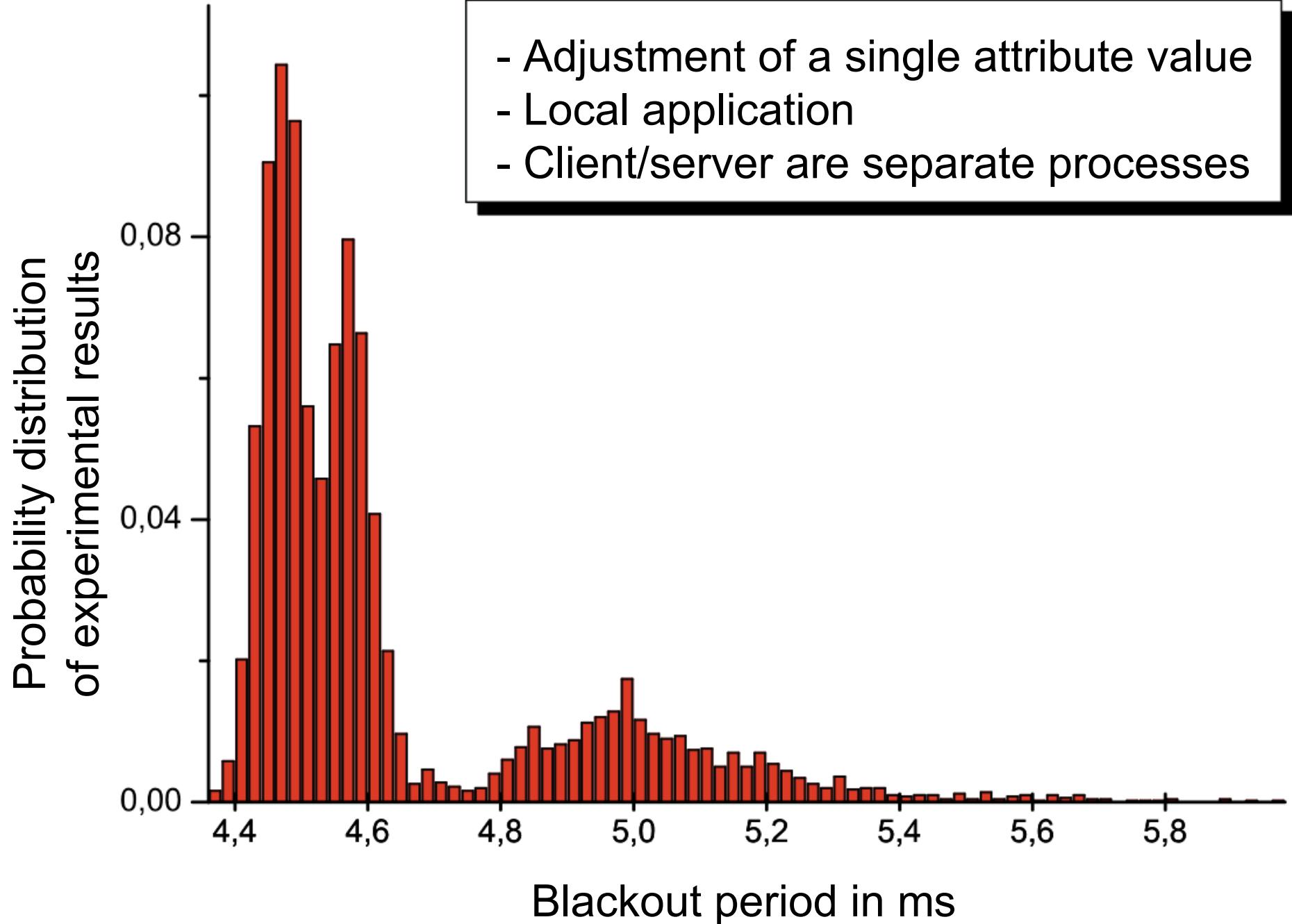
AP 04/03

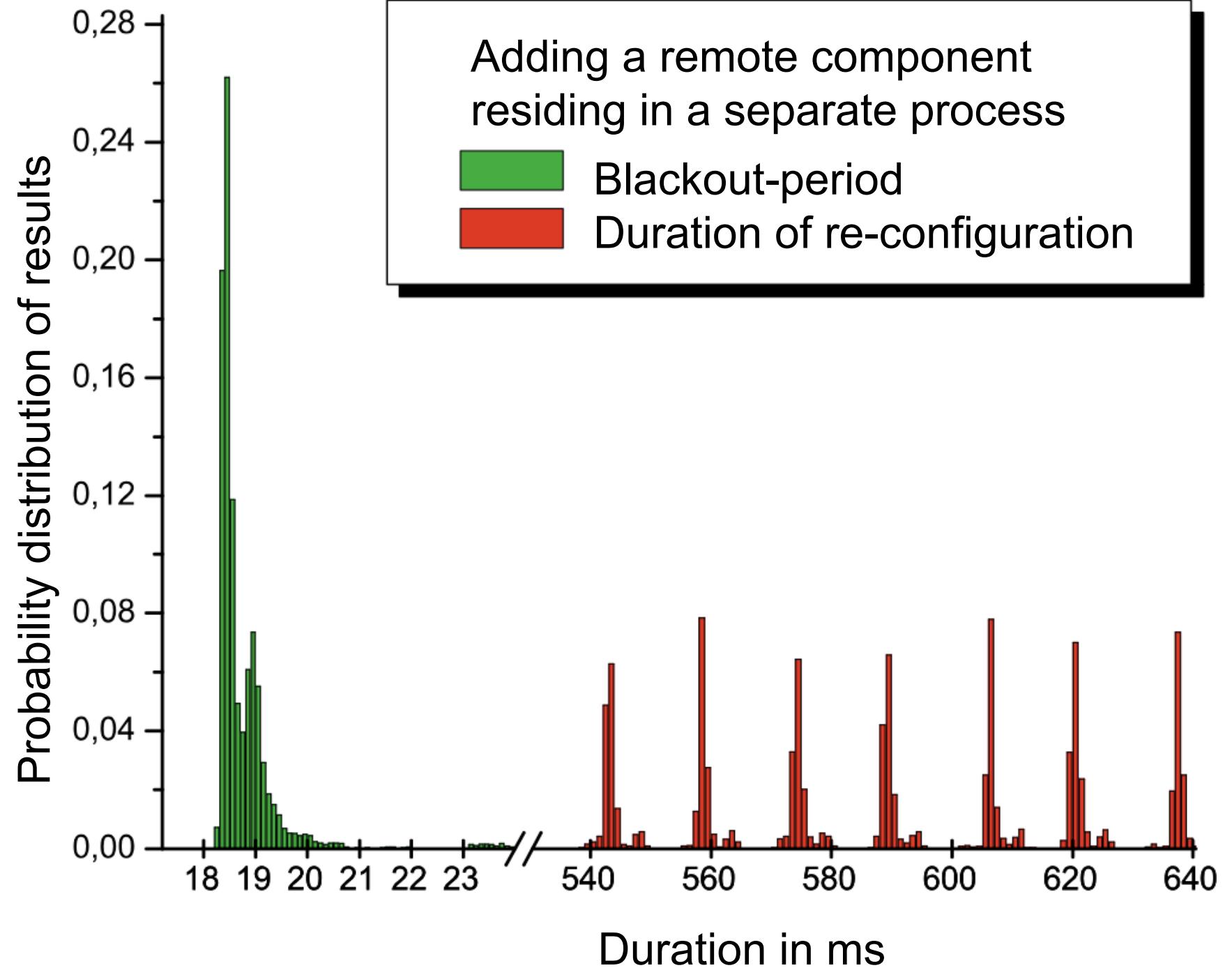
Dynamic Re-Configuration – Experimental Evaluation of .NET

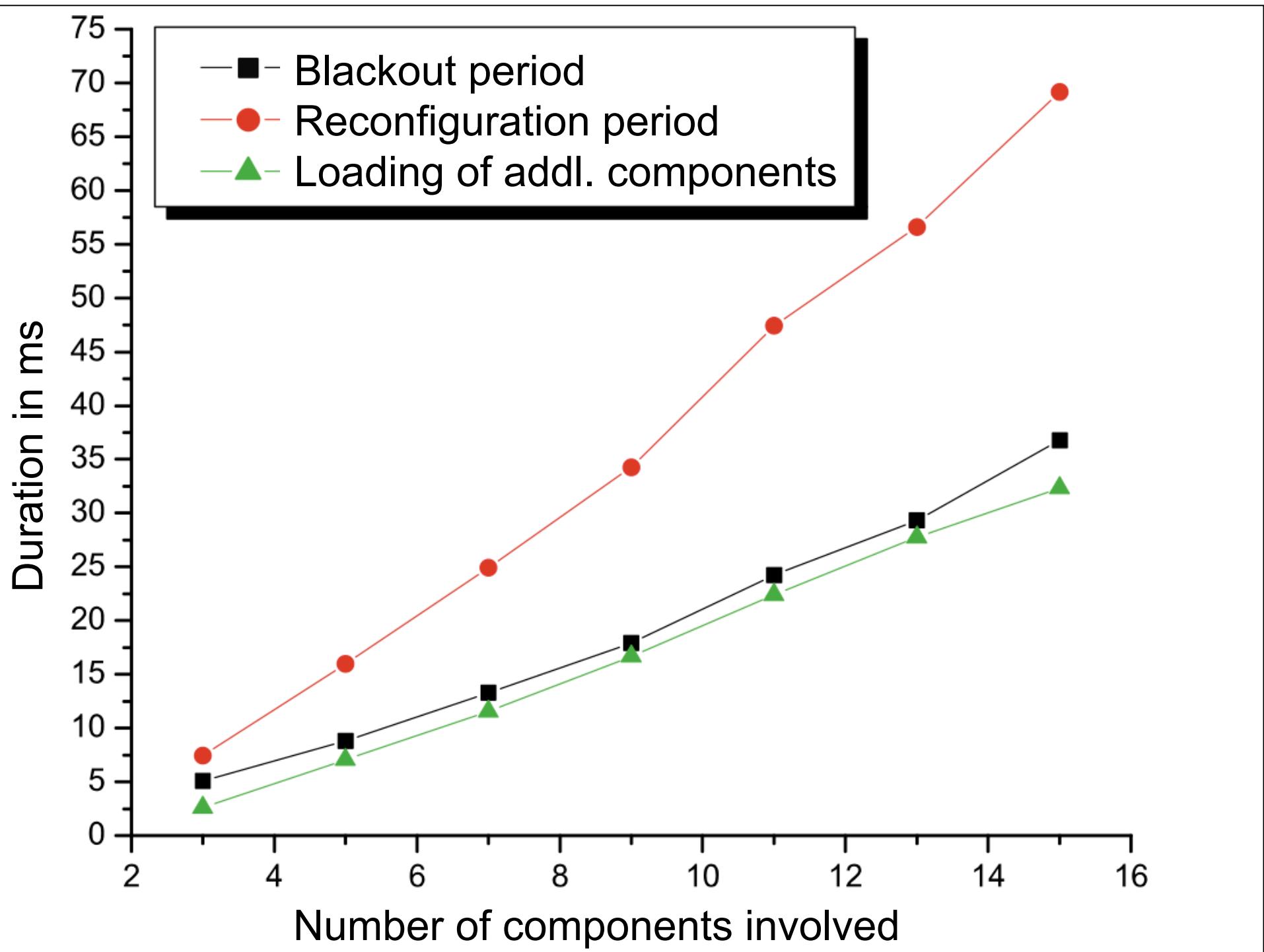
How expensive is dynamic re-configuration?

- Measurements based on Pentium High Resolution Performance Counter (Win32-API)
- Measurement overhead (*unmanaged call*) – within the μ -second range
- Example application: Client/Server system
 - Server propagates attribute via get-method
- 5000-10000 measurements per experiment









Foucault's Pendulum

Web-Experiment - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back → Search Favorites Media Stop Links

Address http://www2.dcl.hpi.uni-potsdam.de/DCLWebInterface/Index.aspx?exp=tcp://was:8005/PendelControlObj&video=wascam.html&expType=Reales-Pendel

Distributed Control Lab

Reales Pendel

Start Stop

Example 1 Example 2 Example 3

Your program has finished because :
Time expired.
The result of the execution of your program follows :

```
runtime of magnet      : 55,553248 s  
resulting speed       : 0,838349457587901 m/s  
runtime of your program : 300,047 s
```

Done Internet

