2.– 5. September 2013
in Nürnberg

Herbstcampus

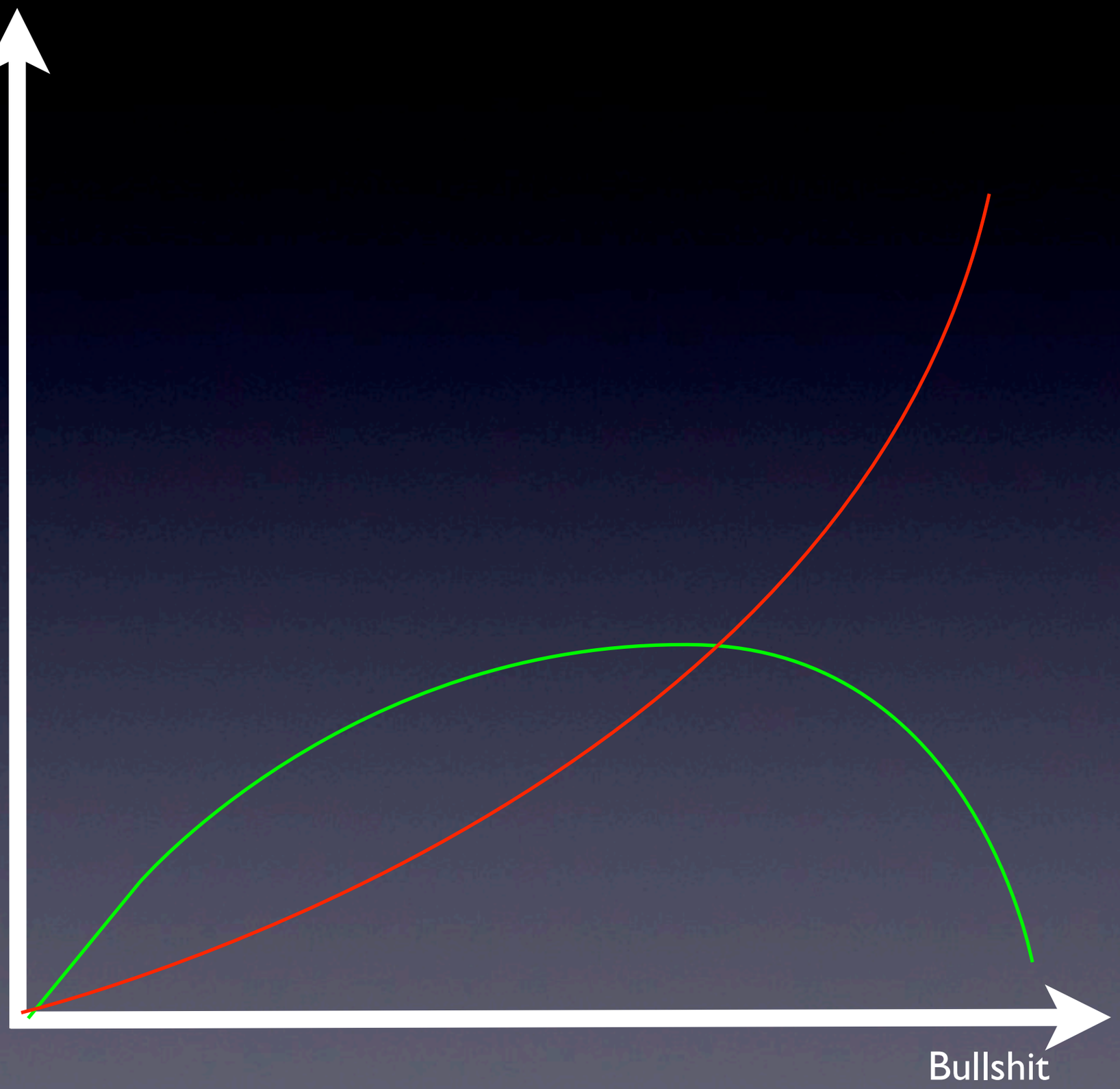
Wissenstransfer
par excellence

Flaschenhälse eliminieren

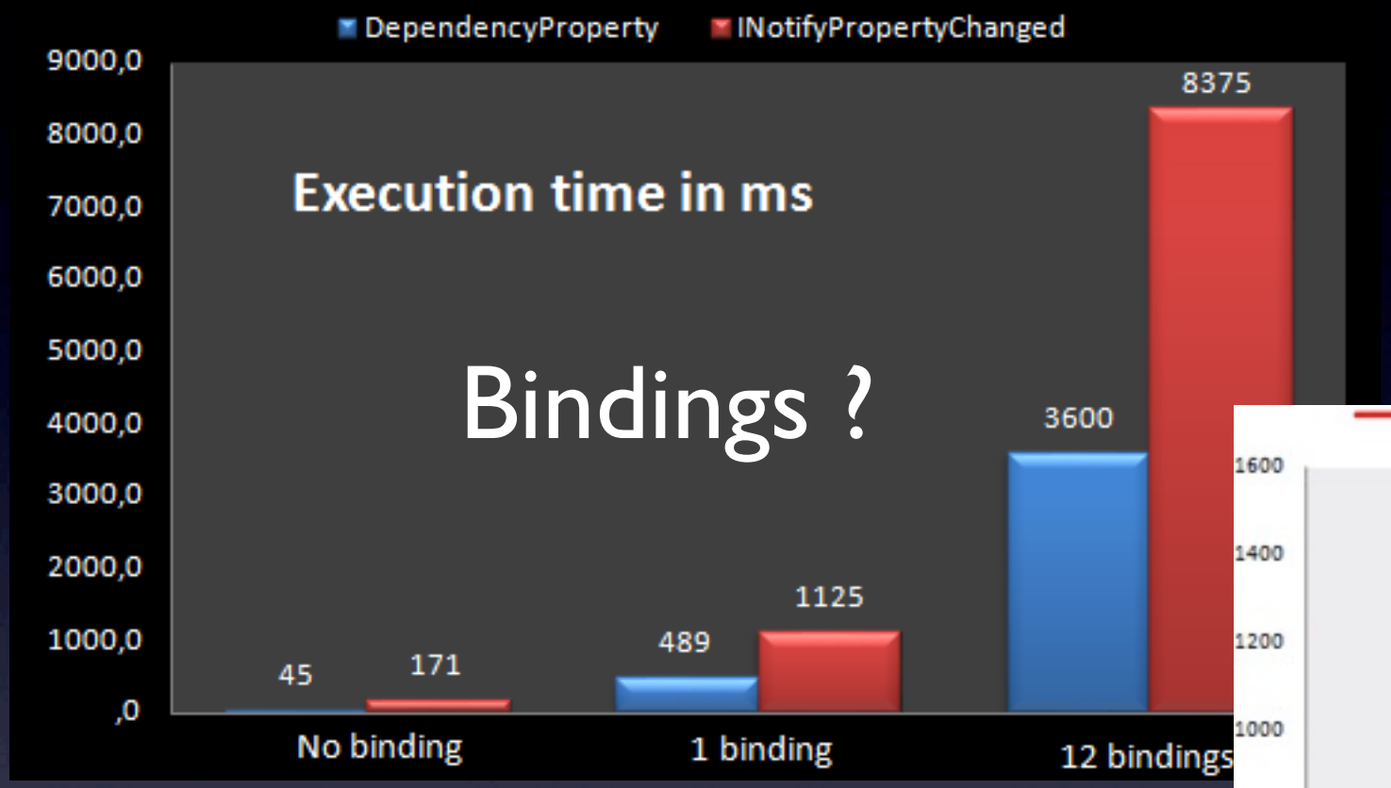
Performance Optimierung von .NET Applikationen

Markus Müller

Datainfo.de

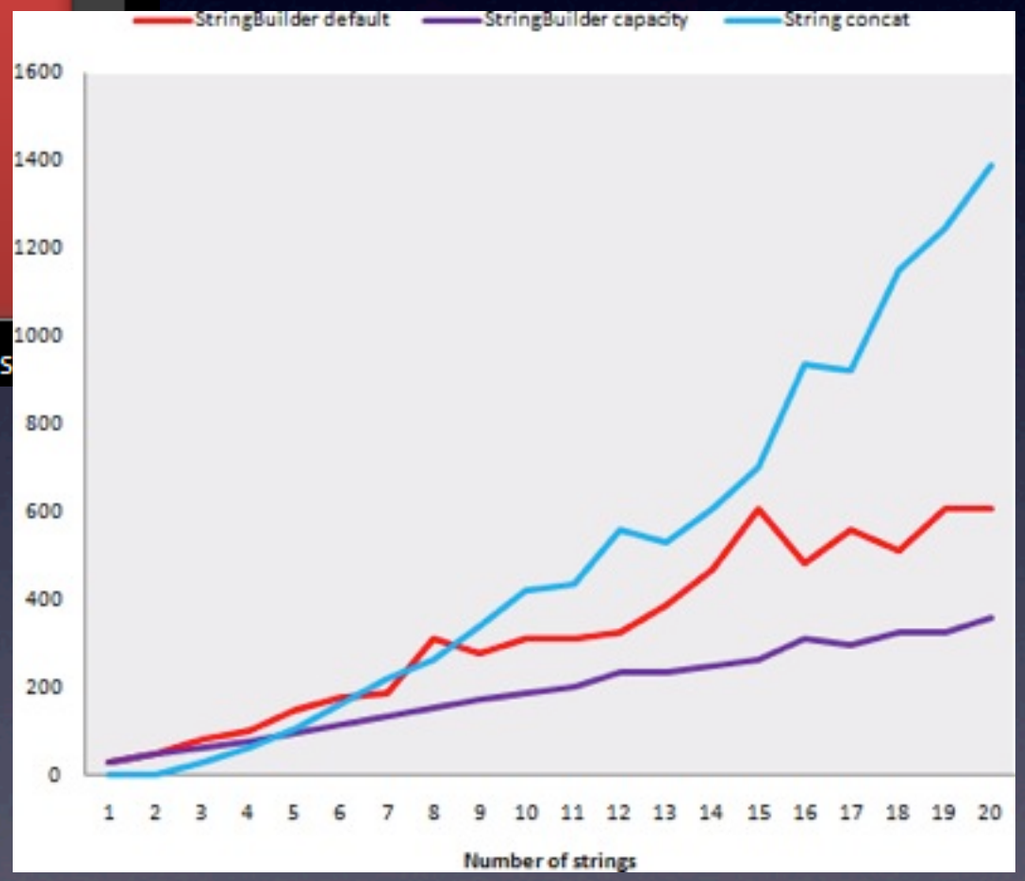

Schmerz
Bullshit

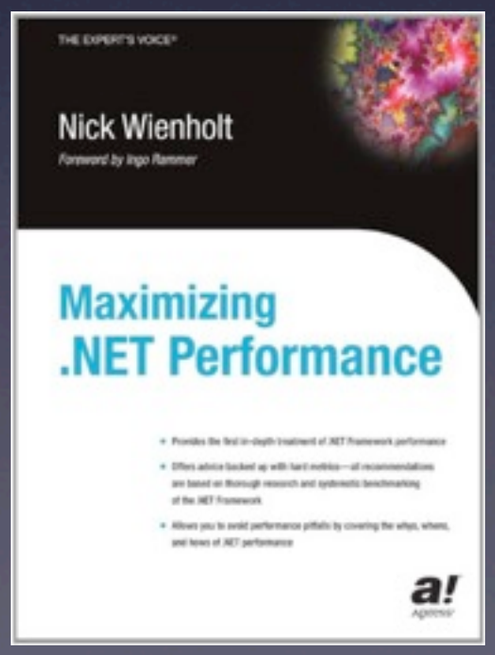
Tweaks.... ????? - not today !!!



Exceptions ?

SQL ?





Security?

Boxing ?

StringBuilder ?

- Rechner werden immer leistungsfähiger
- Stromsparende CPU´s, Speicher etc.
- Schnelle SSD statt Festplatte

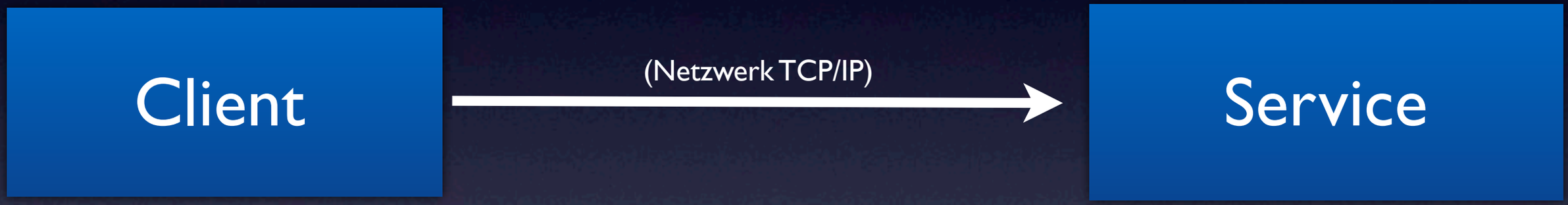
Green Software ?

Natürliche Performance- Feinde moderner Software

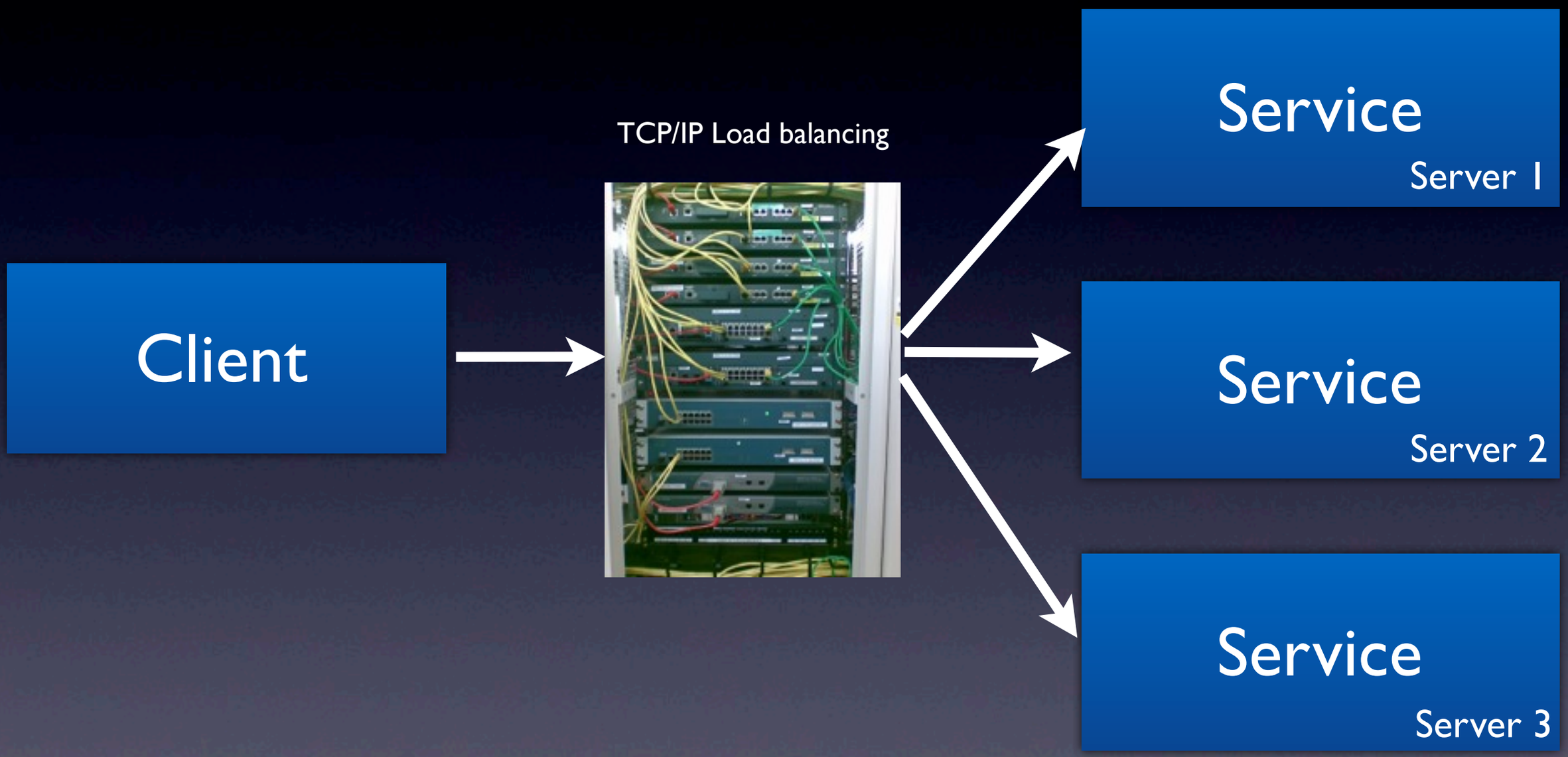
- Zeit / Qualität / Wachsende Anforderungen
- Architektur (Struktur, Abstraktion)
- Frameworks (DI, Generierte Events, PlugIns, Aspekte)

- Horizontale Skalierung

Service



Horizontal skalierbar



Windows Azure

datainfo azure

DASHBOARD MONITOR CONFIGURE **SCALE** INSTANCES LINKED RESOURCES CERTIFICATES

PRODUCTION STAGING

datainfo-azure

roles

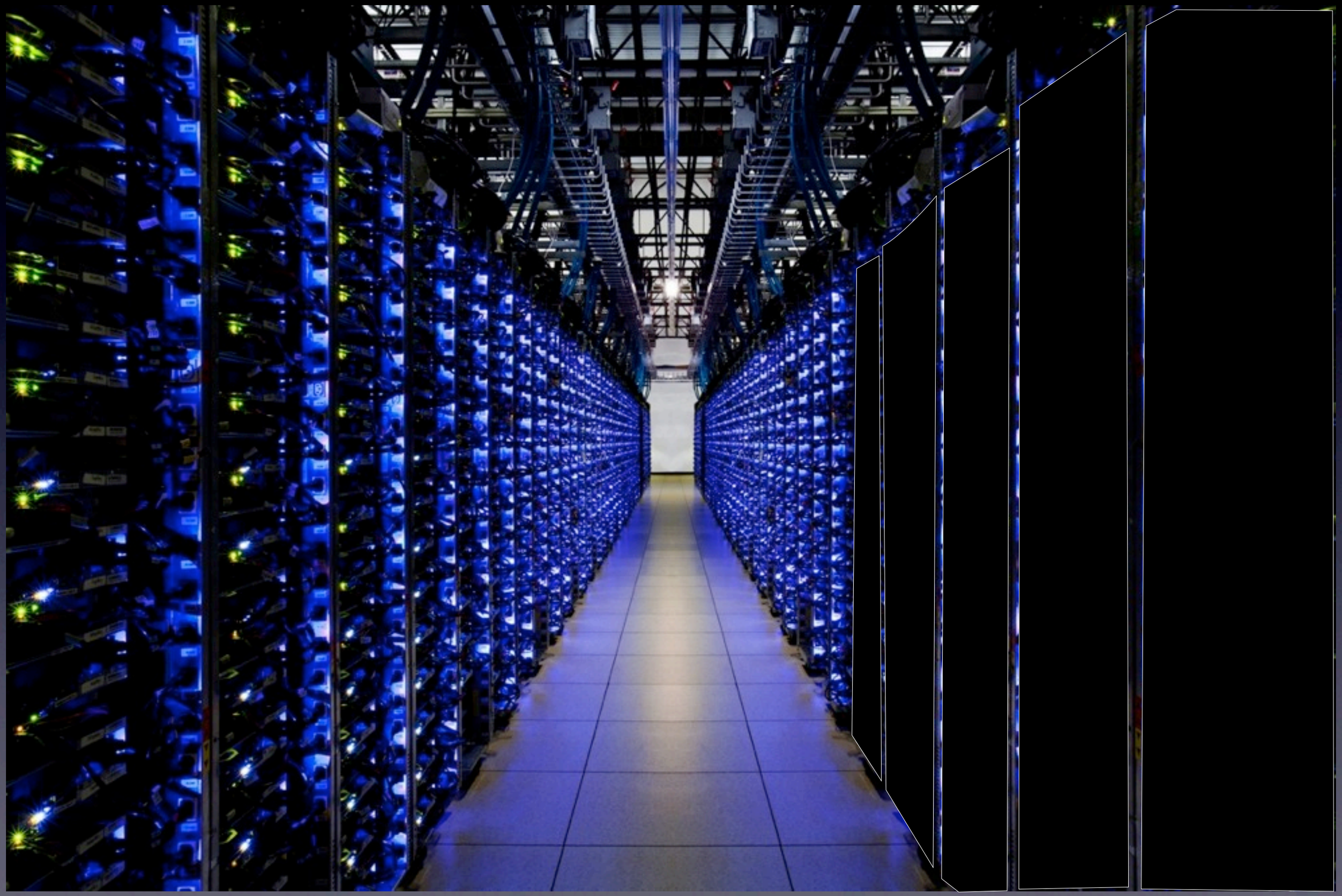
HELLOWORLD_WEBROLE
SMALL (1 CORE, 1.75 GB MEMORY)
VIRTUAL MACHINE

6 instance(s)

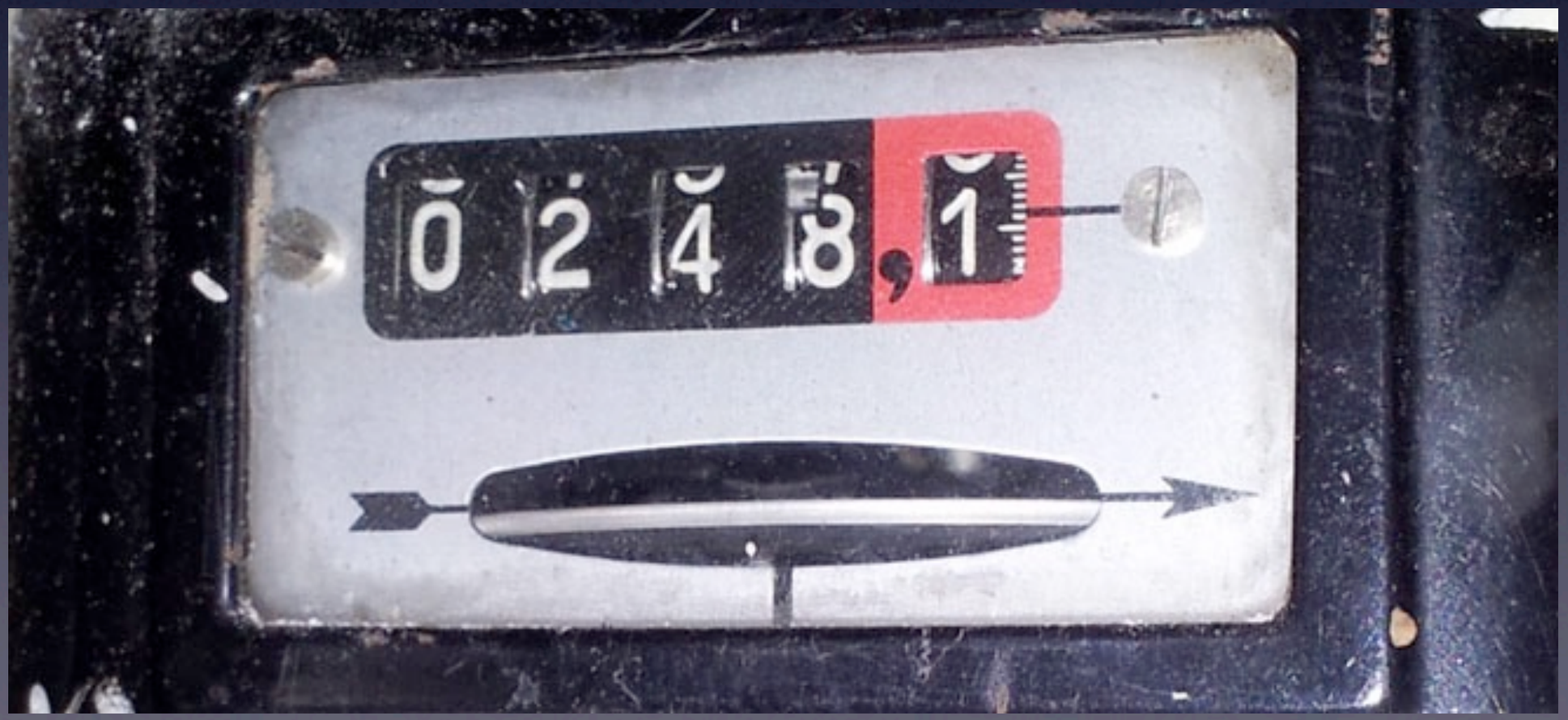
CORES USED BY ROLE CORES USED BY ALL AVAILABLE CORES

NEW SAVE DISCARD 1

Just add some servers....

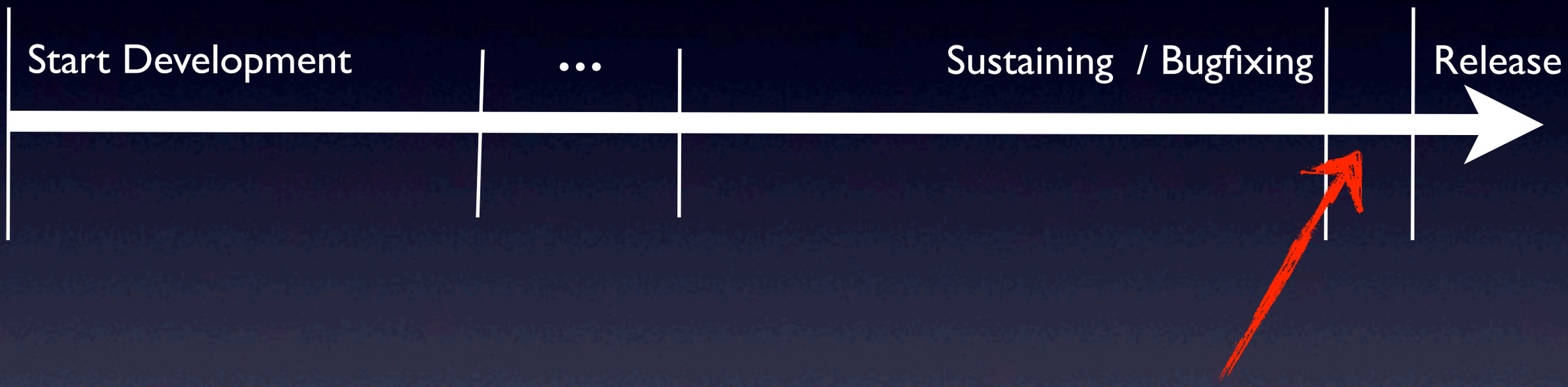


„Wieviele kWh können wir sparen wenn wir unseren Service optimieren?“



Produktmanagement / Customer

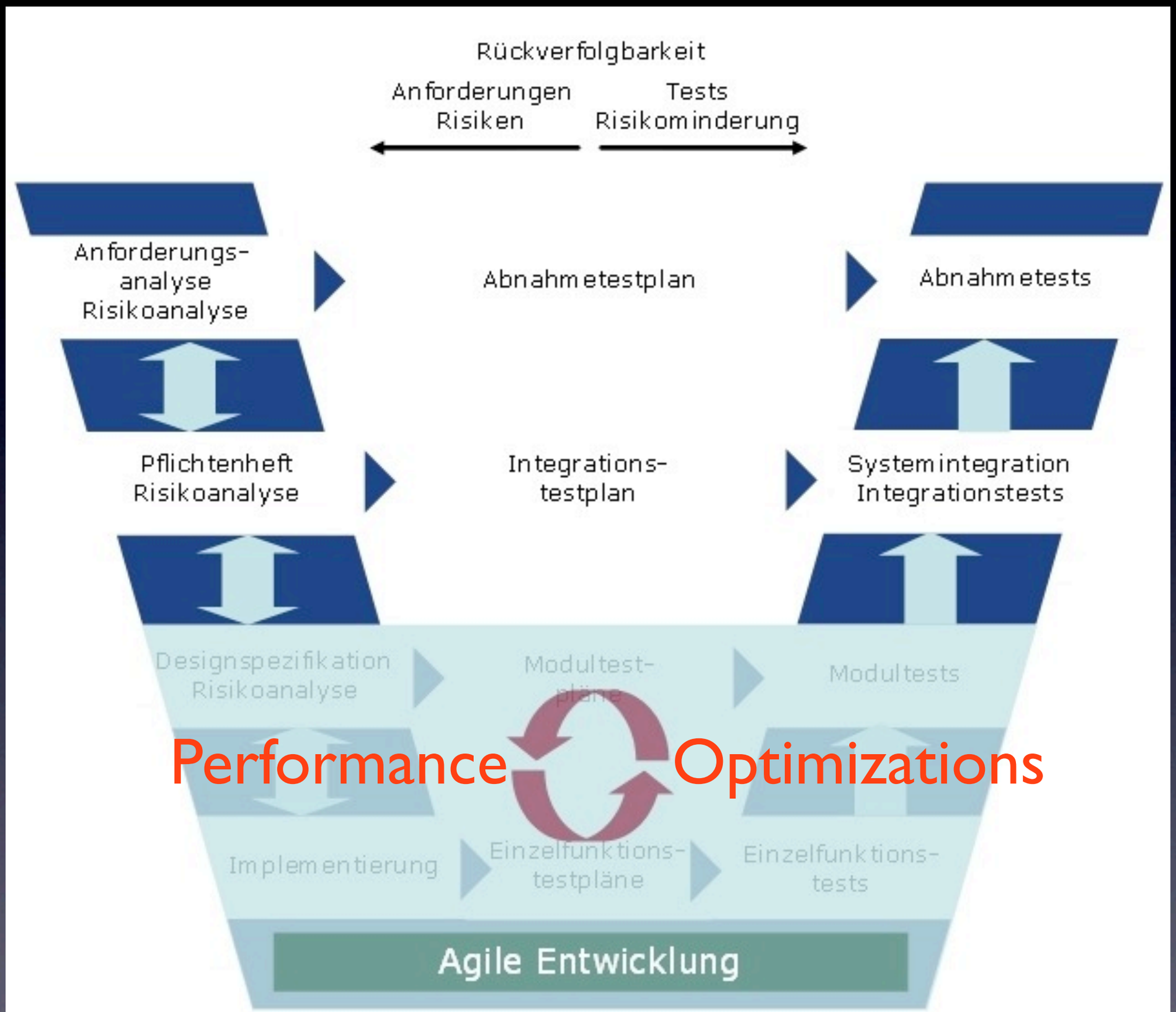
Entwicklungsprozess ?

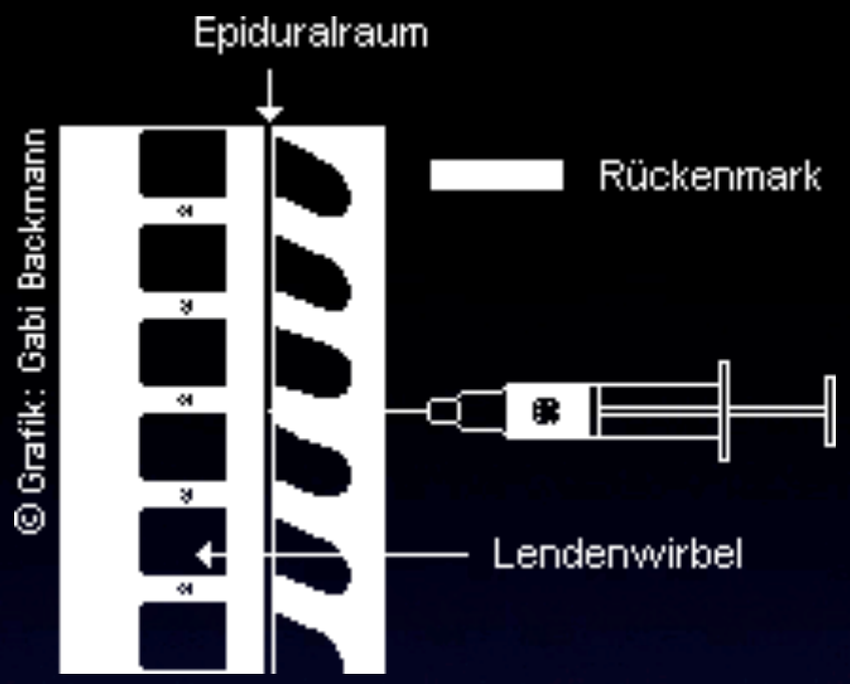


„Klassische“ Performance-Optimierung

„Einfach drauf los entwickeln“

Einbau von „Traces“


Rückverfolgbarkeit
Anforderungen
Risiken
Tests
Risikominderung
Anforderungs-
analyse
Risikoanalyse
Abnahmetestplan
Abnahmetests
Pflichtenheft
Risikoanalyse
Integrations-
testplan
Systemintegration
Integrationstests
Designspezifikation
Risikoanalyse
Modultest-
pläne
Modultests
Performance
Optimizations
Implementierung
Einzelfunktions-
testpläne
Einzelfunktions-
tests
Agile Entwicklung



PDA

Performance Driven Architecture

Performance Prozess

- Definieren von exakten use cases
  (ISO 9126 / non functional requirements)
- Bestandteil des iterativen Prozesses
- Analyse der use cases (Profiling)
- Dokumentieren und Erfassen
  als Defect / WorkItem (Bug tracking system)

Tools, Tools, Tools...

BMC Software AppSight

AQtime Pro
by SMARTBEAR

dynaTrace
software

GlowCode
World's Fastest Profiler

EQATEC
Application Analytics

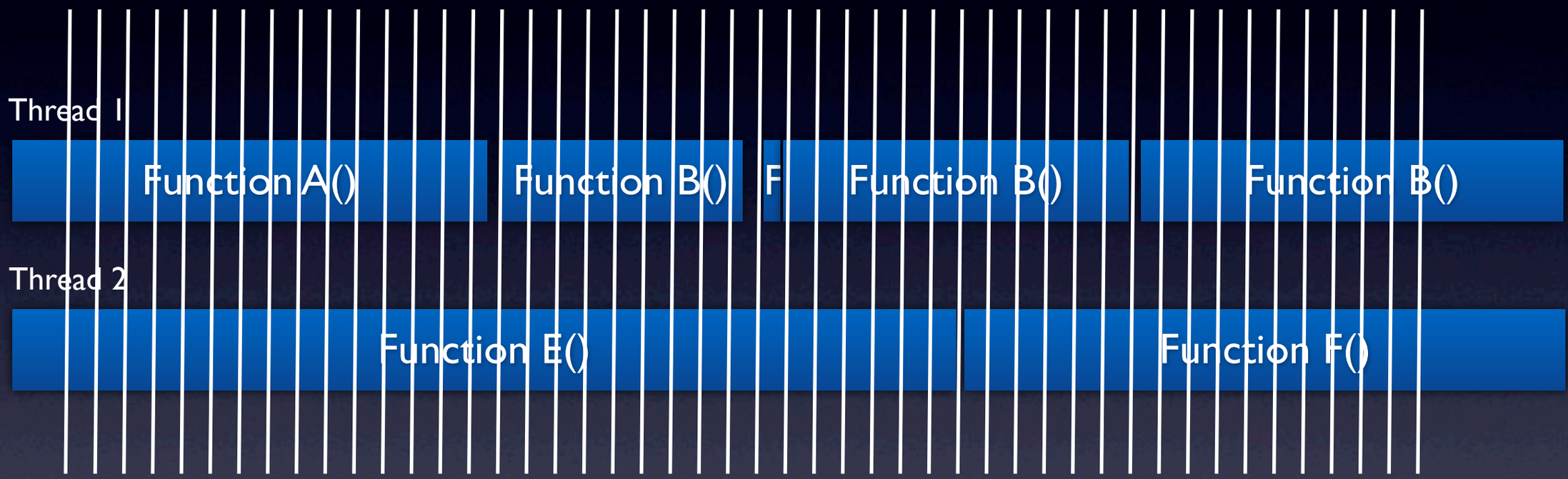
ANTS
PERFORMANCE
PROFILER

PER ∞
performance is our passion

Profiling

Sampling

Zeit



Sampling

- Vorteile :
  + Schnell - geringer Systemeinfluss
  + Prozessübergreifend alle Module sichtbar
- Nachteile :
  - Zeitraster oft zu grob
  - Anzahl der Funktionsaufrufe nicht sichtbar

Visual Studio

Mandel070914.vsp | Start Page

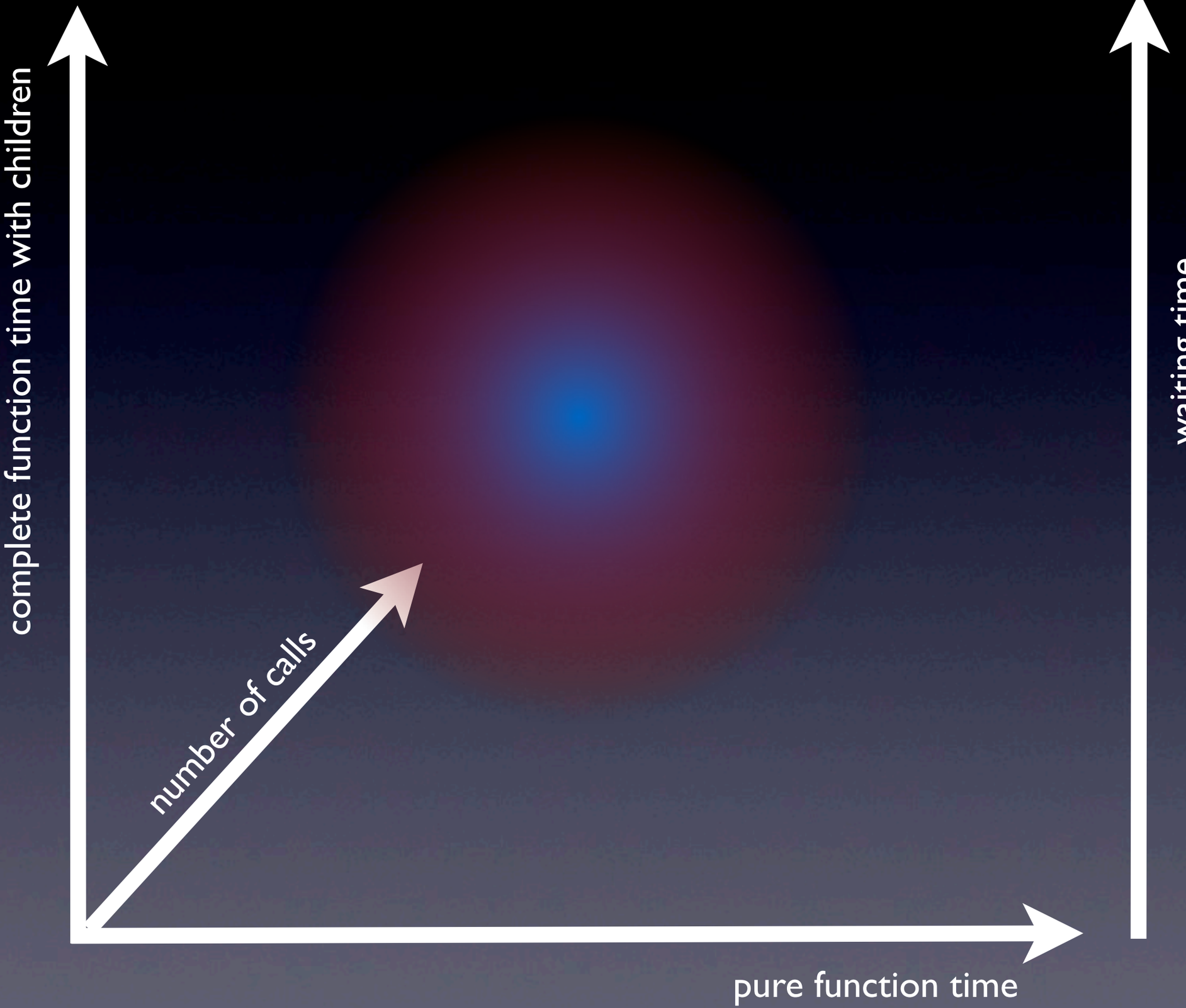
Current View: Call Tree

Function Name	Inclusive Samples	Exclusive Samples
Mandel.exe	5,916	0
Thread::intermediateThreadProc(void *)	4,617	0
Unknown Frame(s)	87	0
SVR::GCHeap::GarbageCollect(int,int,int)	4,530	0
ManagedThreadBase::KickOff(struct ADID,void (*)(void *),void *)	4,530	0
MethodTable::CallFinalizer(class Object *)	4,530	0
Thread::InternalReset(int,int,int,int)	4,530	0
Thread::ShouldChangeAbortToUnload(class Frame *,class Frame *)	4,530	0
WKS::CallFinalizer(class Object *)	4,530	0
ThreadNative::KickOffThread(void *)	4,530	0
MethodDescCallSite::CallWithValueTypes_RetArgSlot(unsigned __int	4,530	0
MethodDesc::CallTargetWorker(unsigned char const *,class Met	4,530	0
_CallDescrWorker@20	4,530	0
MethodDescCallSite::MethodDescCallSite(class MethodDe	4,530	0
__CT??_R0PAVEEMessageException@@@84	4,530	0
ThreadNative::IsAlive(class ThreadBaseObject *)	2	2
System.Threading.ThreadHelper.ThreadStart()	4,528	0
System.Threading.ExecutionContext.Run(cla	4,526	0
System.Threading.ThreadHelper.Thread:	4,526	0
Mandel.Form1.WaitForThread()	651	11
Mandel.Form1.DrawMandel()	3,875	1,958
System.Drawing.Bitmap.SetPixel	1,905	17
System.Drawing.Bitmap..ctor(int32,i	1	0
Mandel.Form1.GetColors(string)	3	0
CLRStubOrUnknownAddress	8	3
CLRStubOrUnknownAddress	2	0
__CorExeMain@0	1,299	0

Instrumentation

- Vorteile :
  + Hoher Informationsgehalt (Anzahl Aufrufer)
  + Sehr genau - Erfassung aller Funktionen
- Nachteile :
  - Systemeinfluss mit wachsender Modulanzahl
  - Instrumentierung notwendig (Build)
  - Signierte Assemblies nicht sichtbar

Finding the Bottleneck



Performance Profiling

- Nur relevante use case aufzeichnen
- Mehrere Aufzeichnungen durchführen
- Keep target platform in focus !!!
- VM Profiling vermeiden

Demo : Mandelbrot

C64 Rechenzeit Grundform : ca 25 Minuten

```
FM 205 PRINT "{CLEAR}"
PJ 210 FOR J = 0 TO HEIGHT - 1
EA 220 : FOR I = 0 TO WIDTH - 1
DR 230 :   GOSUB 300
RR 231 :   GOSUB 400
CP 240 : NEXT I
JG 250 NEXT J
SE 290 GET A$:IF A$ = "" THEN 290
MM 299 END
FM 300 REM CALCULATE MANDELBROT
BC 310 ISMND = -1
PF 312 NREAL = XL + I * XINC
RK 313 NIMG  = YL + J * YINC
XS 315 RZ = 0:IZ = 0
RS 316 R2Z = 0:I2Z = 0
EQ 320 FOR K = 1 TO REPS
CC 330 : R2Z = RZ*RZ - IZ*IZ
JQ 340 : I2Z  = 2*RZ*IZ
QJ 350 : RZ   = R2Z+NREAL
HX 360 : IZ    = I2Z+NIMG
EE 370 : IF (RZ*RZ + IZ*IZ)>4 THEN ISMND=0:K=REPS
QA 390 NEXT K
GB 399 RETURN
GB 400 REM PLOT MANDELBROT
GE 410 COUNT = I+J*WIDTH
PJ 420 POKE 1024+COUNT,160
GR 430 POKE 55296+COUNT,2 AND NOT ISMND
BH 499 RETURN
```

Speed Trace Pro

Some Key Features:

- Minimal intrusion/ low impact
- Distributed Tracing
- CPU Timeline
- Information clustering, Decompile etc.

2.– 5. September 2013
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Markus Müller

Datainfo.de