

We Are Reactive

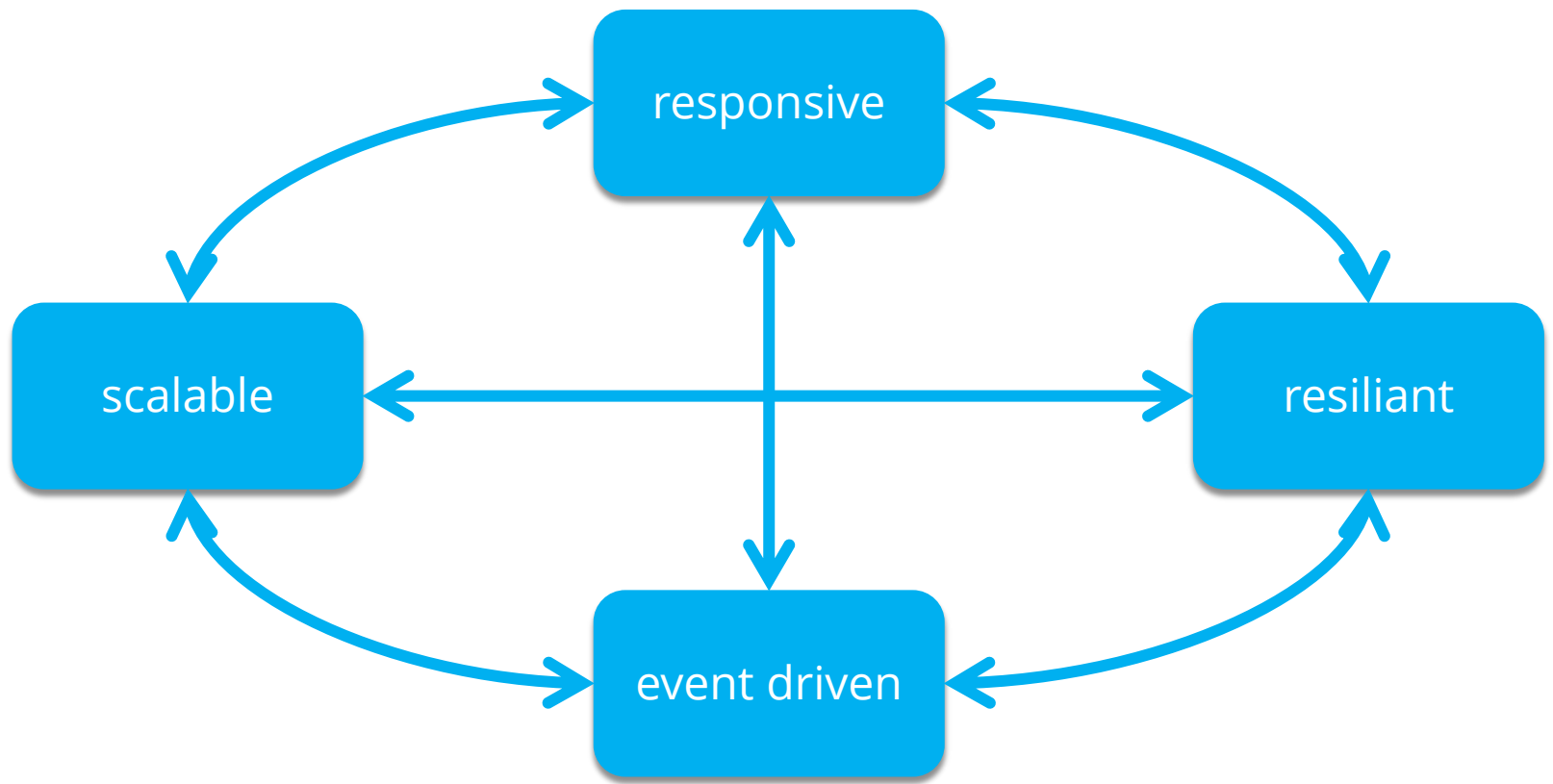
# REACTIVITY

Niko Köbler & Heiko Spinder  
[www.Qualitects-Group.com](http://www.Qualitects-Group.com)

# Was ist Reaktivität?

- Reaktives Manifest  
[www.reactivemanifesto.org](http://www.reactivemanifesto.org)
- Beschreibt die 100% ideale Welt
- Kann nur Hilfestellung sein

# Was ist Reaktivität?



# Bekannte Konzepte

- Enterprise-Architektur:
  - JMS
  - CEP
  - SOA
- Technische Herausforderungen in den Griff bekommen
- Konzepte halten Einzug in die Applikation / auf Ebene der Programmierung

# DEMO

JavaScript – Meteor

Java – Akka

# Meteor

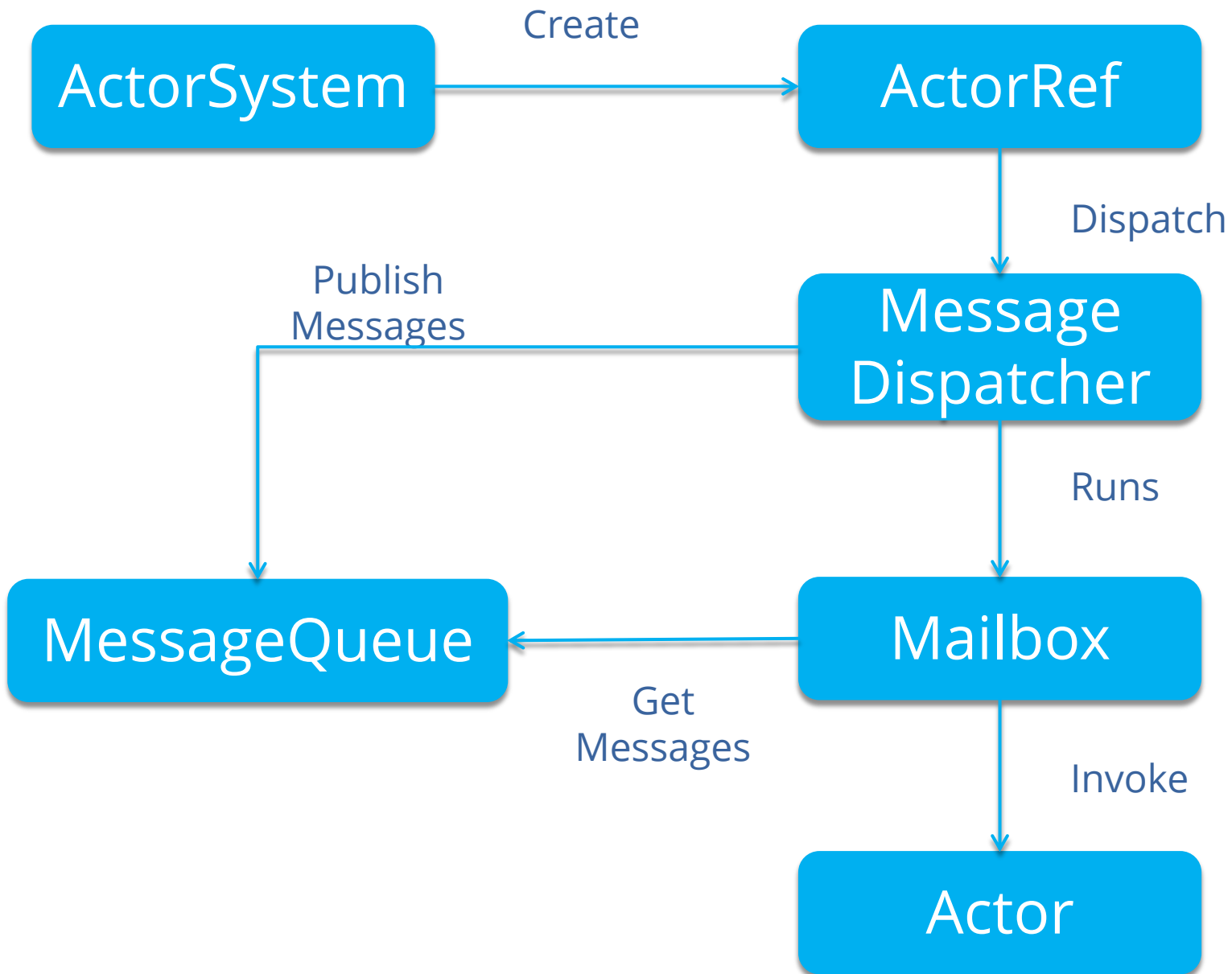
- <http://www.meteor.com>
- JavaScript Fullstack Plattform (DB bis UI)
- Features
  - ▣ JavaScript
  - ▣ Daten, kein HTML
  - ▣ „Überall-Datenbank“
  - ▣ Daten-Synchronisierung
  - ▣ Latenz Kompensation

# Akka



We Are Reactive

- <http://akka.io>
- Implementierung des Actor-Modells mit Scala und Java
- Features
  - ▣ Actors tauschen asynchrone Nachrichten
  - ▣ Lose Kopplung
  - ▣ Actors sind leichtgewichtig
  - ▣ Distribution
  - ▣ Fehlerbehandlung





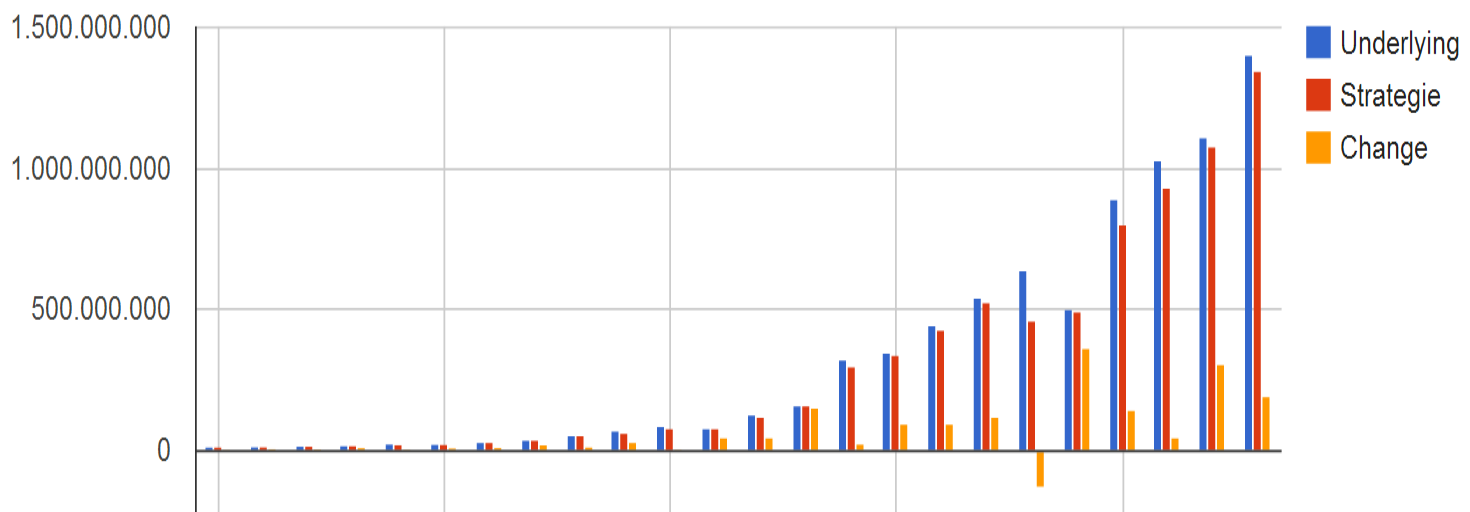
<a href="#">globalbest_Pop11.rule (Link)</a>	9.274.678,00	27.11.1990	26.06.2013	Dax_D1_OLHC_1991_2013
<a href="#">globalbest_Pop2.rule (Link)</a>	131.616.800,00	27.11.1990	26.06.2013	Dax_D1_OLHC_1991_2013
<b><a href="#">globalbest_Pop3.rule (Link)</a></b>	15.589.016,00	27.11.1990	26.06.2013	Dax_D1_OLHC_1991_2013
<a href="#">globalbest_Pop4.rule (Link)</a>	540.236.608,00	27.11.1990	26.06.2013	Dax_D1_OLHC_1991_2013

## Strategie: globalbest\_Pop3.rule

[Test Json](#) [Nachrichten löschen](#)

[Informationen](#) [Performance](#) [Jahresvergleich](#) [Neue Auswertung](#) [Appinfo](#)

globalbest\_Pop3.rule



# Geänderte Anforderungen

Anforderung	früher		heute
Antwortzeitverhalten	sek	↘	ms
Anzahl Server	1 – 10	↗	100 – 1000
Downtime	Stunden	↘	n/a
Datenvolumen	GB	↗	TB

# Vorteile von Reaktivität

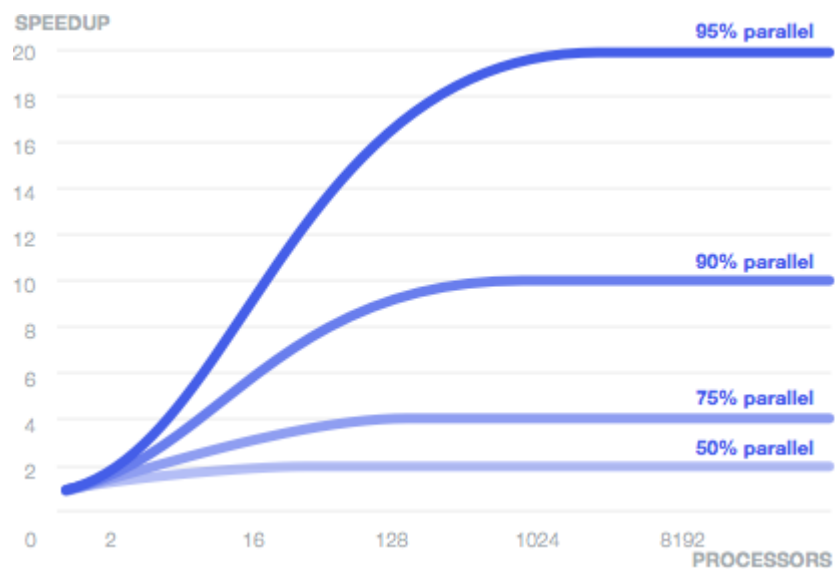
- Teile und herrsche
- Separation of Concerns
- Mehr Möglichkeiten Applikationen zu designen
- Frameworks können unterstützen und Arbeit abnehmen

# Frameworks

- Node.js
- Vert.x
- Akka
- Play
- Reactor
- Meteor
- react.js
- ...

*ABER:  
Ein Framework alleine  
bringt noch keine  
Reaktivität!*

# Amdahlsches Gesetz



The speedup of a program using multiple processors in parallel computing is limited by the sequential fraction of the program. For example, if 95% of the program can be parallelized, the theoretical maximum speedup using parallel computing would be 20× as shown in the diagram, no matter how many processors are used.

*(from: Wikipedia/Reactive Manifesto)*

# Herausforderungen

- Asynchrones / nicht-blockierendes Verhalten
  - macht (zunächst) mehr Aufwand
  - weil es „neu“ ist
  - widerspricht unserem gewohnten imperativen Programmierstil
- Zustandslosigkeit
- Daten-Distribution

# Herausforderungen

- Erhöhter Aufwand in Architektur-Überlegungen
- Motivation für Reaktivität steckt in den Domänen-Anforderungen
- (bekannte) Komplexitätsgrenzen werden überschritten
- Für kleine Projekte nicht unbedingt sinnvoll

# Testen

- Einfacher UND aufwändiger !



We Are Reactive

# Q & A

Niko Köbler & Heiko Spinder  
[www.Qualitects-Group.com](http://www.Qualitects-Group.com)