

# In-Memory Grid Computing und Java EE

Michael Bräuer  
Oracle Deutschland B.V. & Co. KG

# Safe Harbor Statement

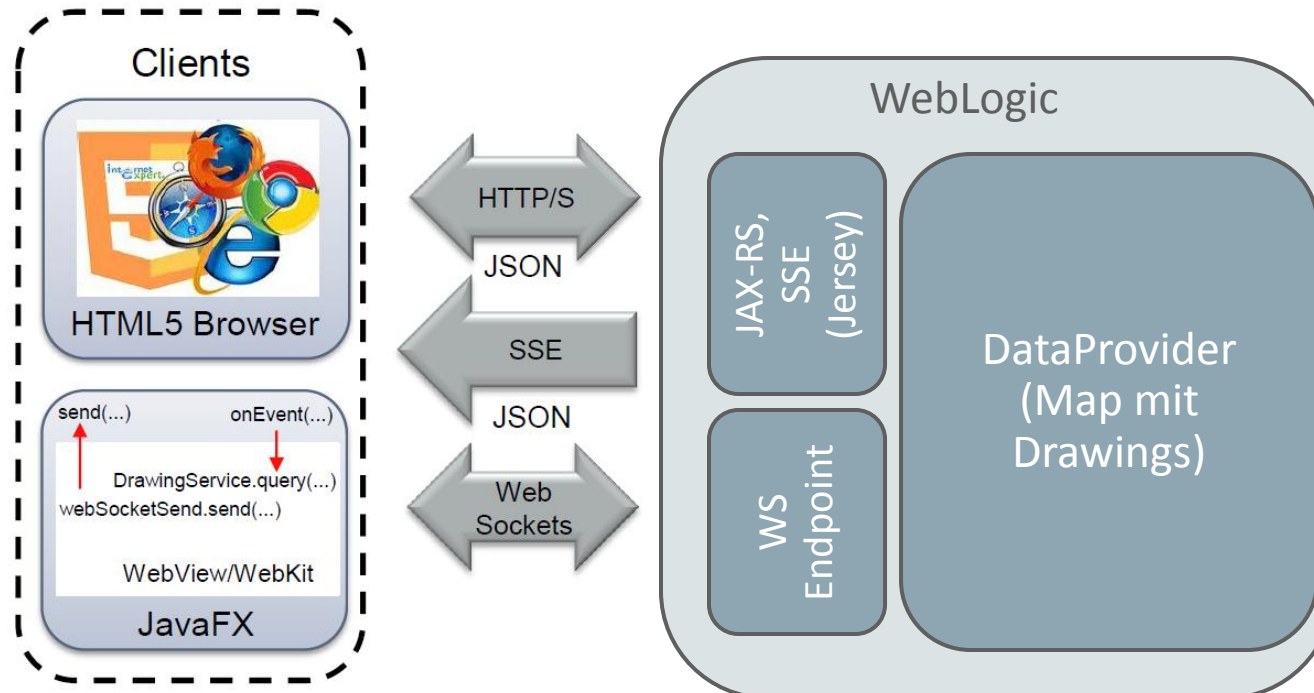
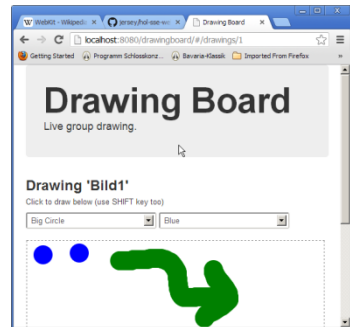
The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Agenda

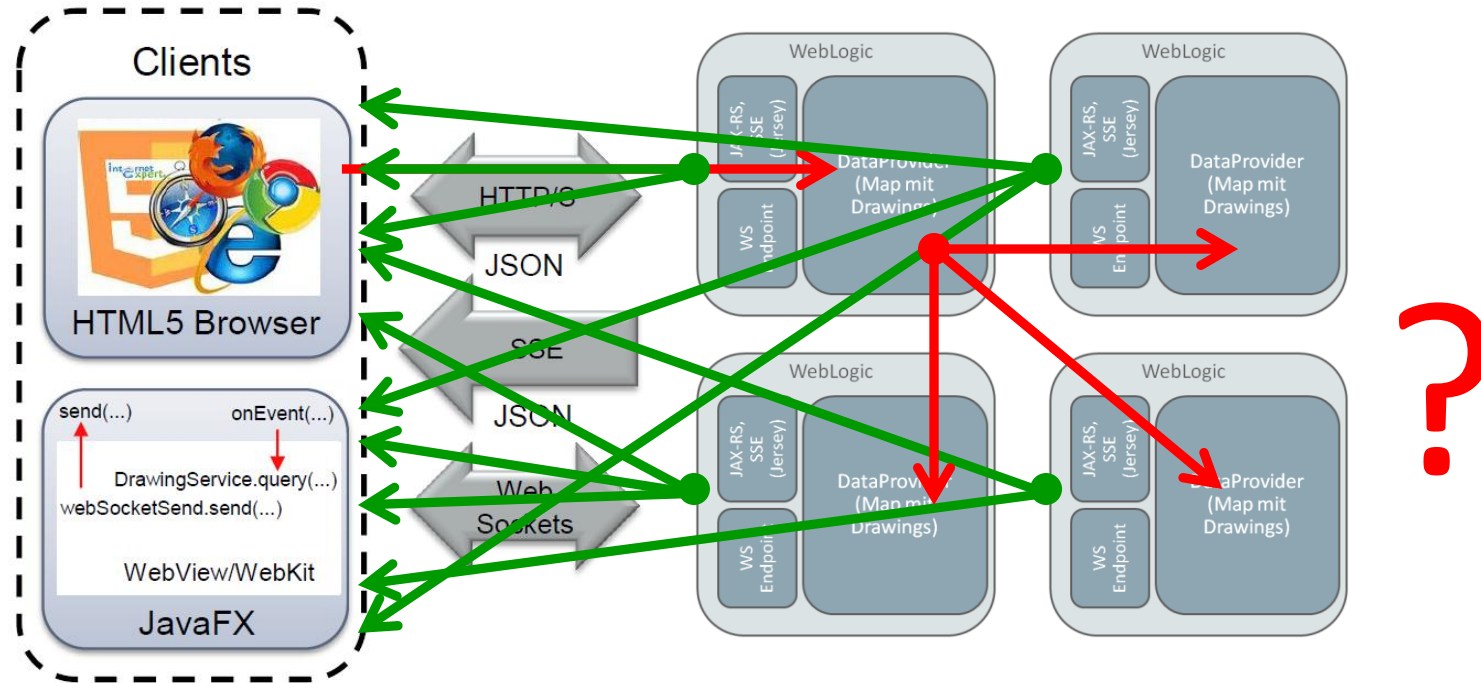
- Status Quo: Zustand in verteilten Java EE Umgebungen
- In-Memory Grid Computing und Java EE mit Oracle Coherence

# Serverseitiger Zustand in modernen Java EE Anwendungen

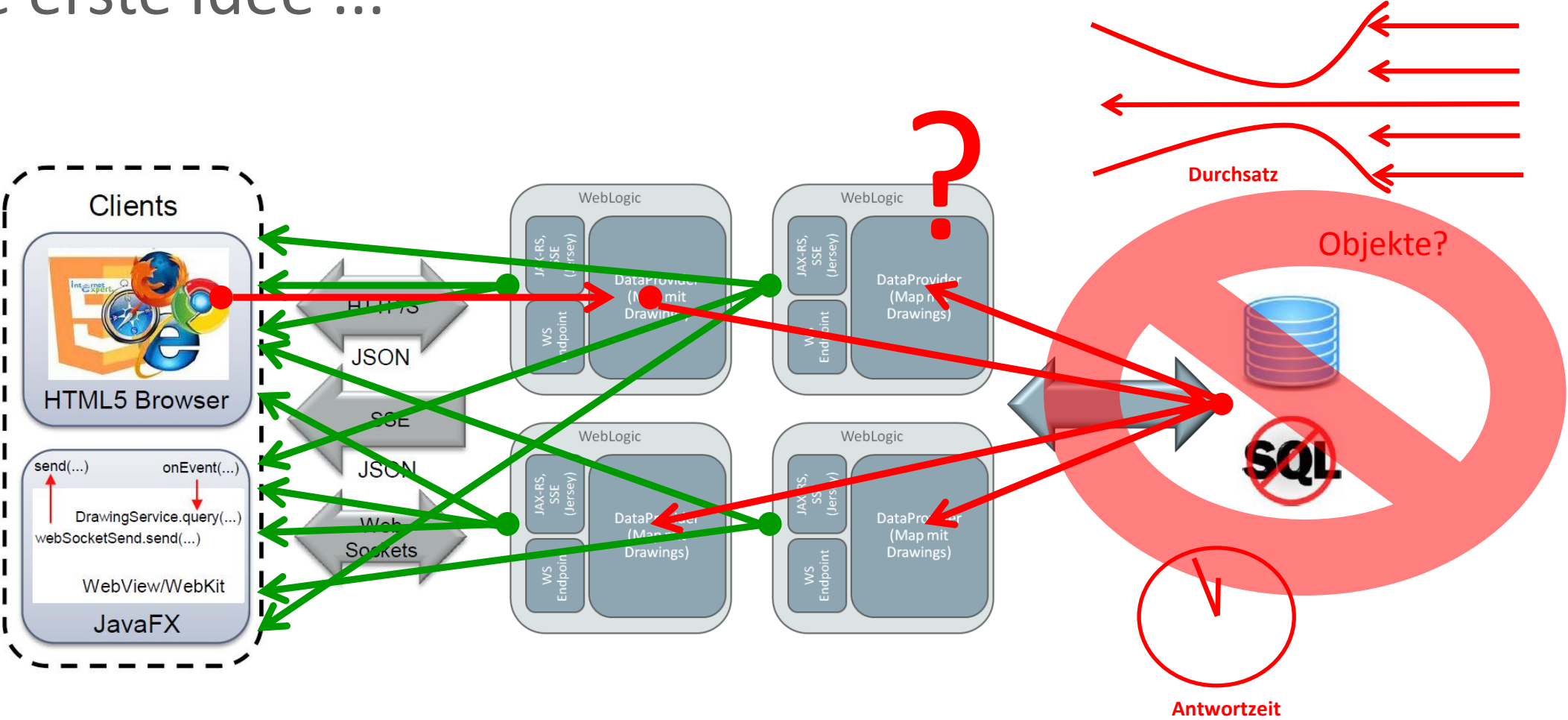
- Beispiel Drawingboard (<https://github.com/doschkinow/hol-sse-websocket>):
  - Client: HTML5 (angular.js)
  - Server: JAX-RS 2.0 (JSR- 339) und SSE, Websockets 1.0 (JSR-356), u.a.

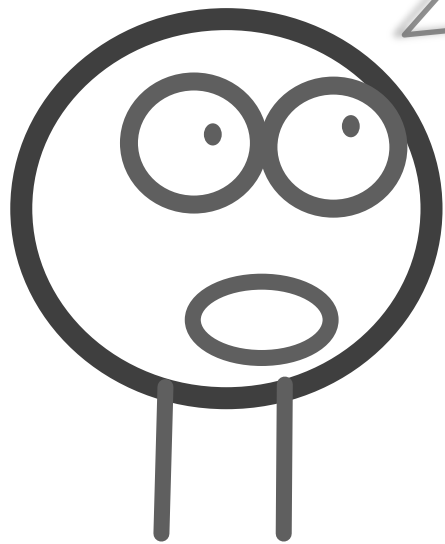


# Zustand, der in verschiedenen JVM gebraucht wird



# Eine erste Idee ...

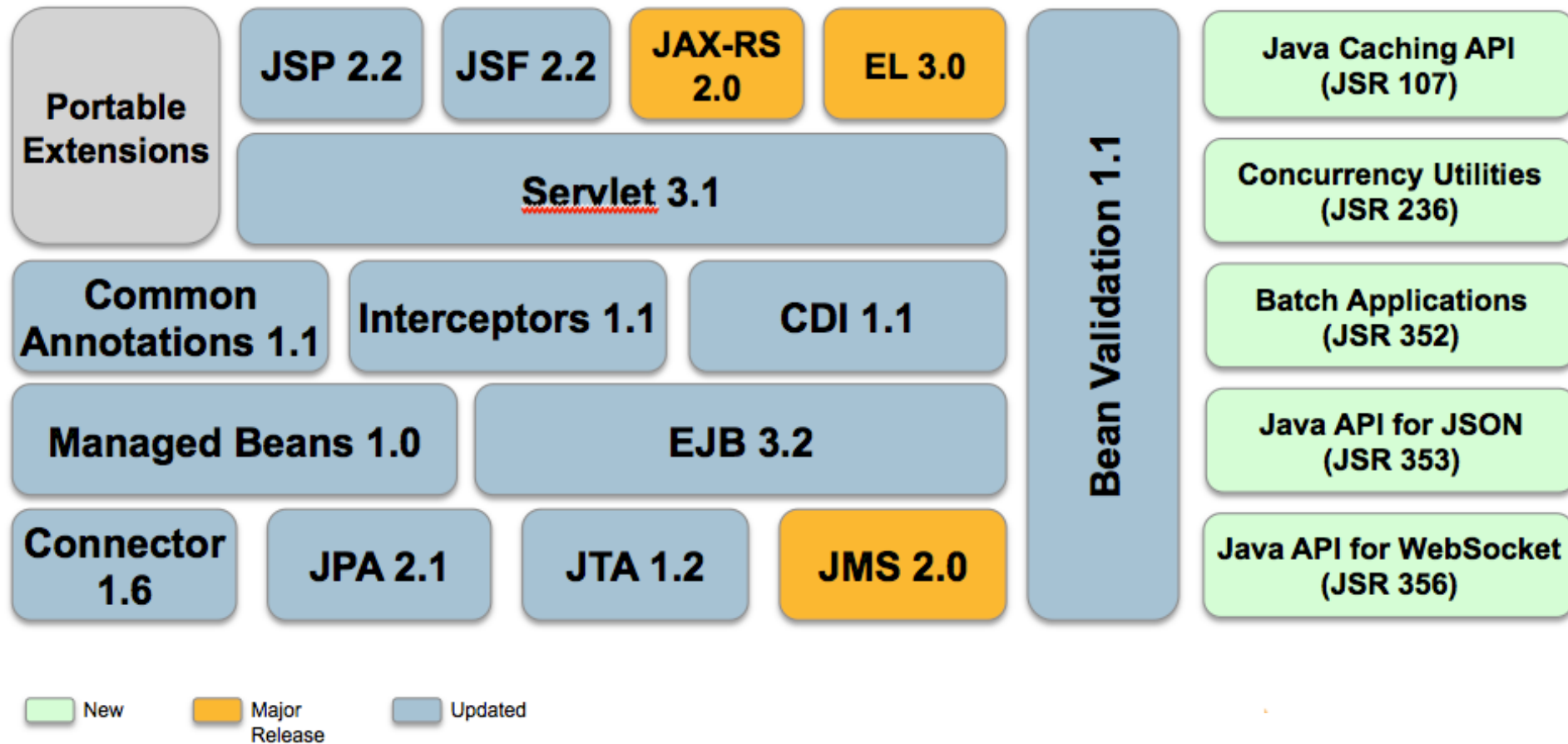




# Java EE

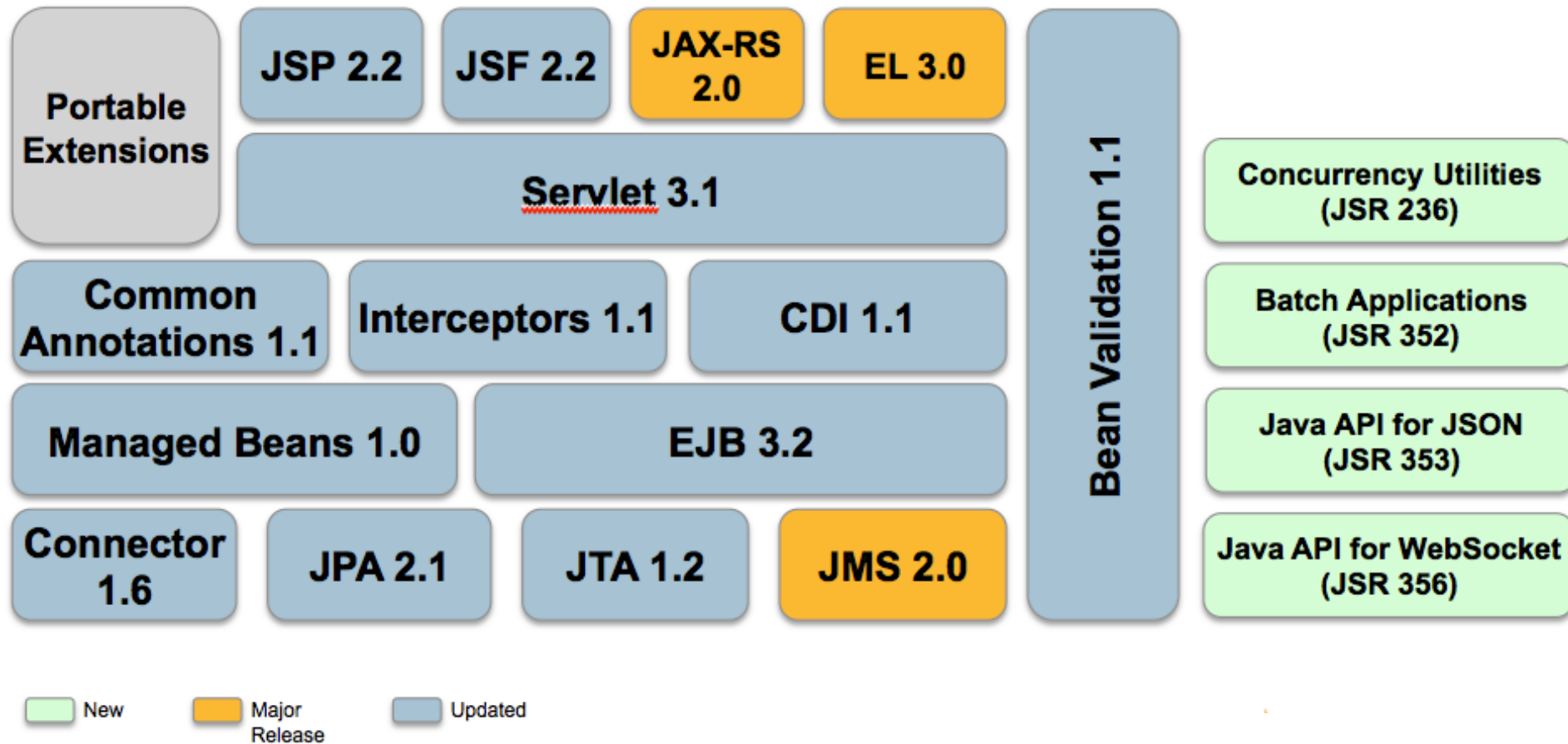
Cached Data ???  
Shared State ???  
Conversational State ???

# Java EE 7 JSRs – Prefinal



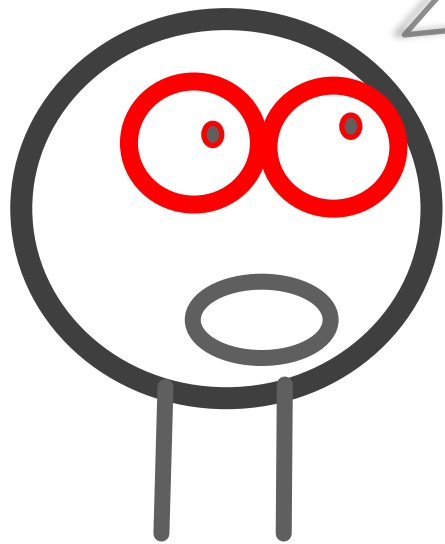


# Java EE 7 JSRs – Final



# Status Quo

- Kein JSR für Caching innerhalb von Java EE
- Kein JSR für verteiltes Grid Computing
- Frühjahr 2014
  - Finalisierung JSR-107: Caching in Java SE
  - Keine Aussagen über Zusammenarbeit mit Java EE → Java EE 8 (?)



Verteilter Zugriff?  
Kapazität?  
Objekte?  
Skalierbarkeit?  
Verfügbarkeit?  
Latenz ?

# In-Memory Data Grid\*

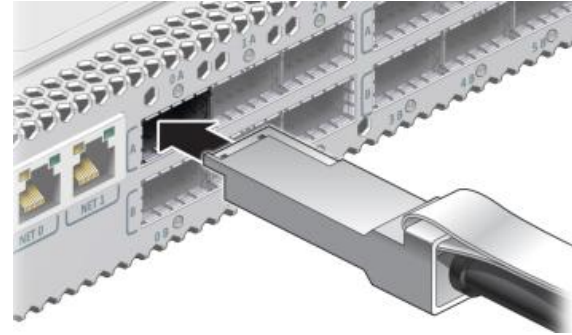
\* angelehnt an [http://www.jroller.com/cpurdy/entry/defining\\_a\\_data\\_grid](http://www.jroller.com/cpurdy/entry/defining_a_data_grid)

- In-Memory basiertes Management System für verteilt genutzte Objekte mit
  - geringen Antwortzeiten
  - hohem Durchsatz
  - vorhersehbarer Skalierbarkeit
  - Fehlertoleranz: Verfügbarkeit und Zuverlässigkeit.

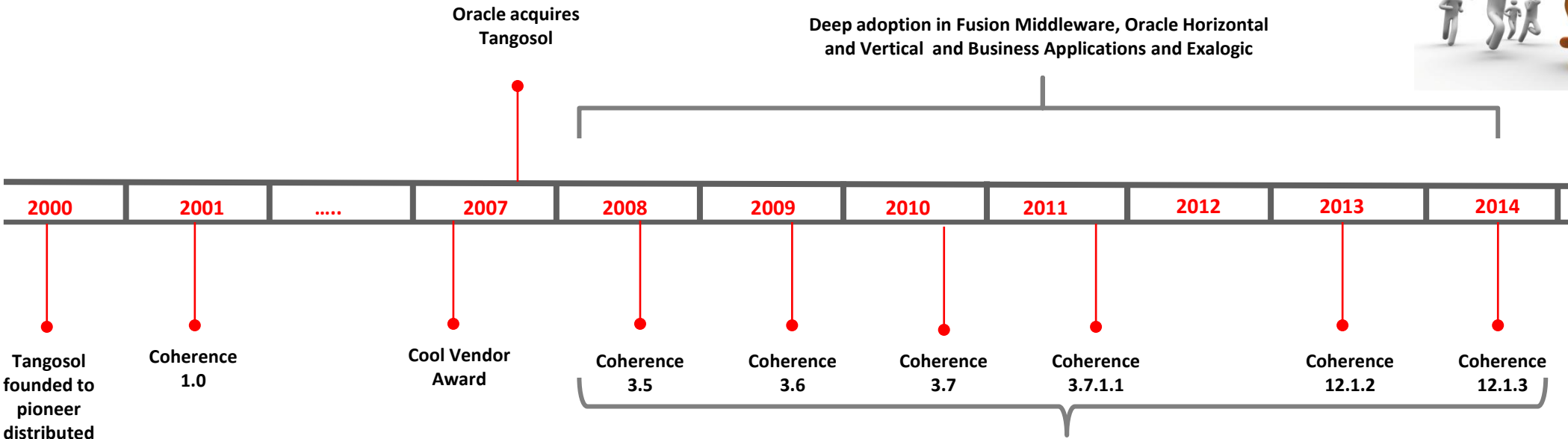


# Anforderungen/Herausforderungen

- Hohe Kapazitäten: TBs
  - In-Memory: On-Heap
  - alternative Medien: SSDs, Flash
  - Schnelles Netzwerk (!!)
- Verteiltes Rechnen: Datenmanagement und parallele, auch asynchrone Verarbeitung, z.B. mittels EntryProcessor, Filter, Aggregationen
- Integration in Java SE und Java EE Umgebungen
- Vielseitiger Clientzugriff: Java, C++, .Net, REST, memcached
- Integration mit persistenten Systemen (Vermeidung Stale Cache Problem)
- Einfache Konfiguration, Administration und Monitoring vieler JVMs



# Pionier im Bereich Java In-Memory Data Grid Computing



Increasing market share and feature capabilities for In Memory Data Grid. Market widened All Categories – Financial Services, E-Commerce, Travel, Insurance, Retail, Manufacturing, Telco ...



# Coherence in a Nutshell

- Coherence-Knoten:
  - Java Knoten:(Non-)Storage Knoten und optional Proxy-Knoten: XML-Konfigurationen und Bibliotheken im Klassenpfad
  - C++, .NET, REST, memcached Zugriff mittels Proxy-Knoten.
- Storageknoten können Daten aufnehmen/verwalten, spezielle Serialisierungsmechanismen sind möglich (POF), Evolvable
- Sehr effiziente Protokolle für Clusterkommunikation (TCMP)
- APIs:
  - Coherence spezifische APIs: NamedCache API (= Map<K,V> ähnliche Datenstruktur)
  - JCache API (JSR-107)

# Coherence in a Nutshell

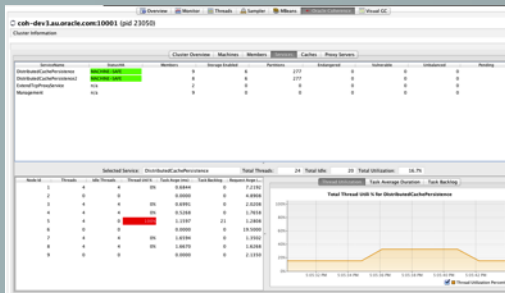
- Alle IDEs
- Builds mit Maven
- Konfiguration over Defaults
  - XML
  - Java System Properties
- Management mit JMX
  - JVisualVM Plug-In
  - Oracle Enterprise Manager Cloud Control und 3rd Party

## Administration vieler JVMs?

### JVisualVM Plugin



- Available now for 3.x on Coherence Community Website
- Lightweight plugin to JVM



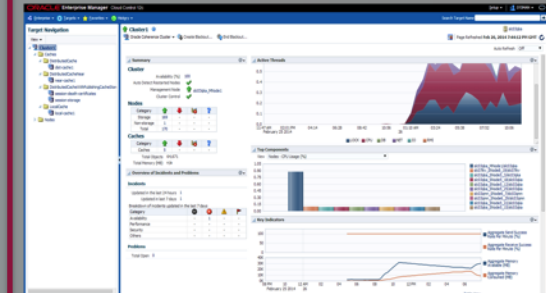
### Fusion Middleware Control

- OOTB administration and monitoring for all FMW
- Dev/QA point-in-time insight into cluster



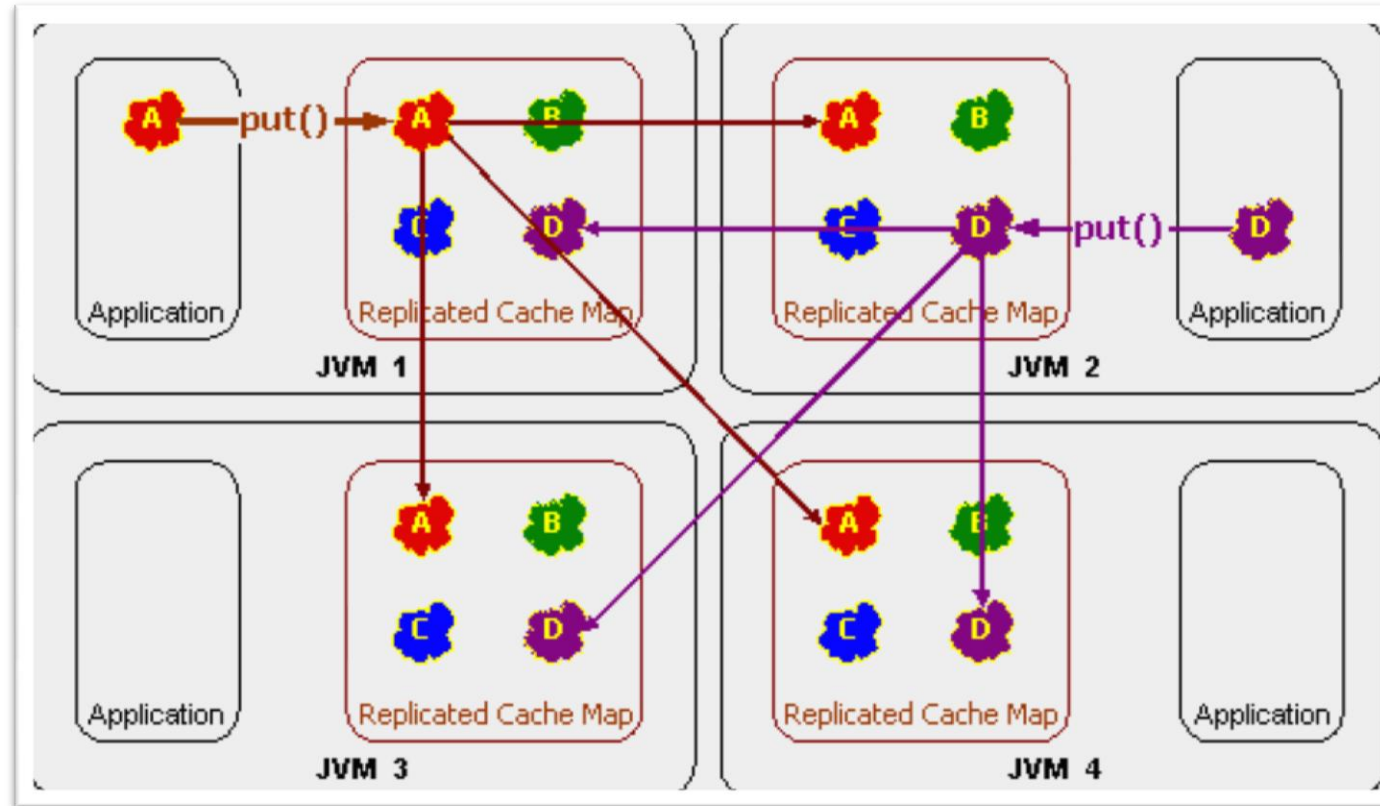
### Coherence Management Pack for OEM

- Complete management and monitoring solution
- Store historical results
- Java diagnostics tooling

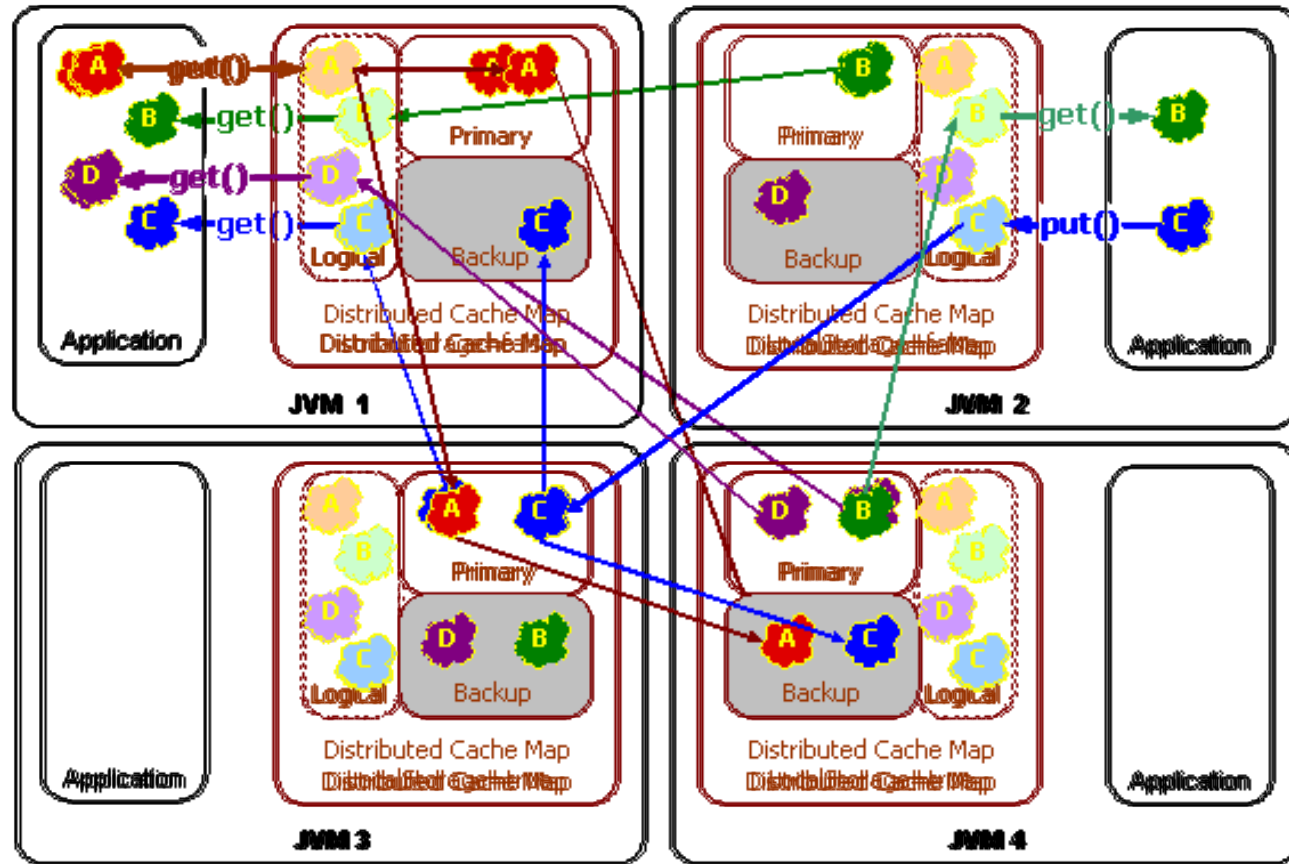




# Coherence in a Nutshell

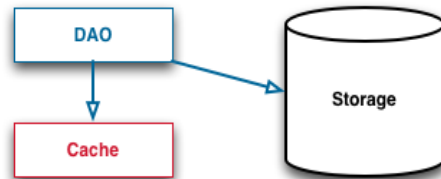


# Coherence in a Nutshell

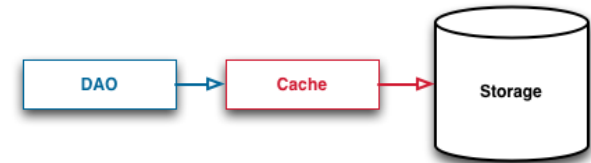


# Coherence in a Nutshell

Cache Aside

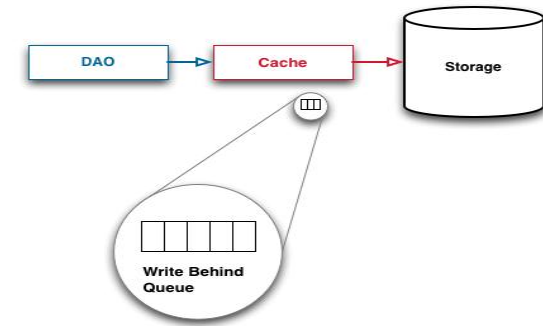


Write-Through

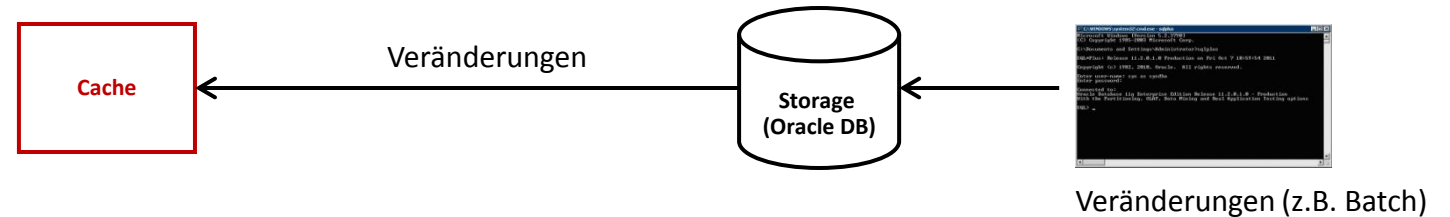


CacheStore

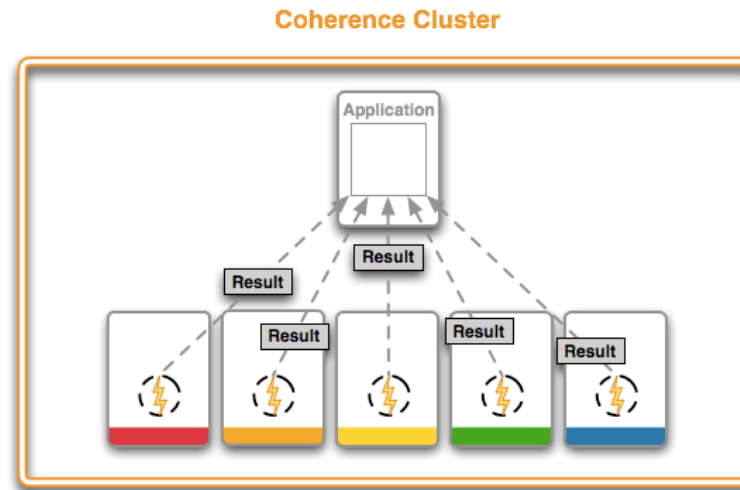
Write-Behind



Coherence GoldenGate HotCache



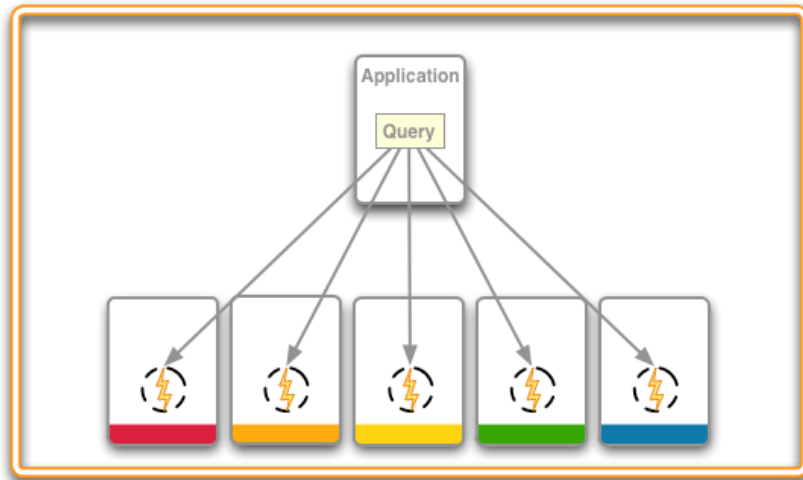
# Coherence in a Nutshell



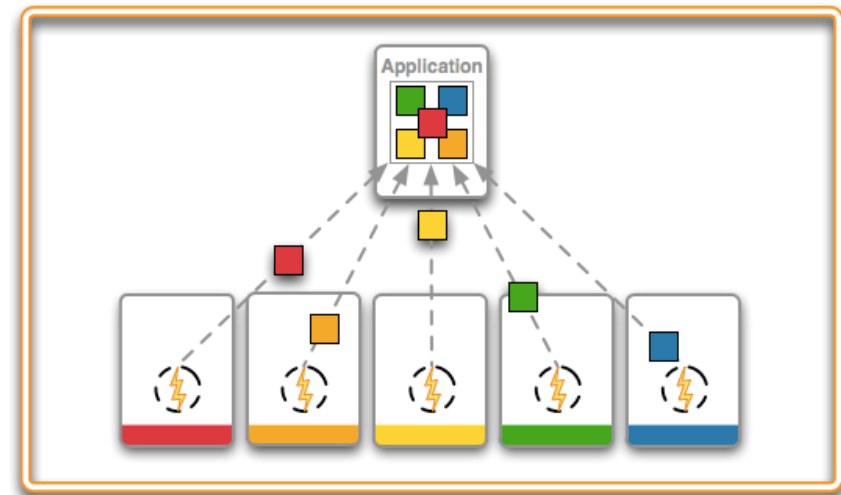
Parallele Verarbeitung: synchrone und asynchrone Entryprozessoren  
Bringe Code zu Verarbeitungsknoten und nicht Daten in den Client

# Coherence in a Nutshell

Coherence Cluster

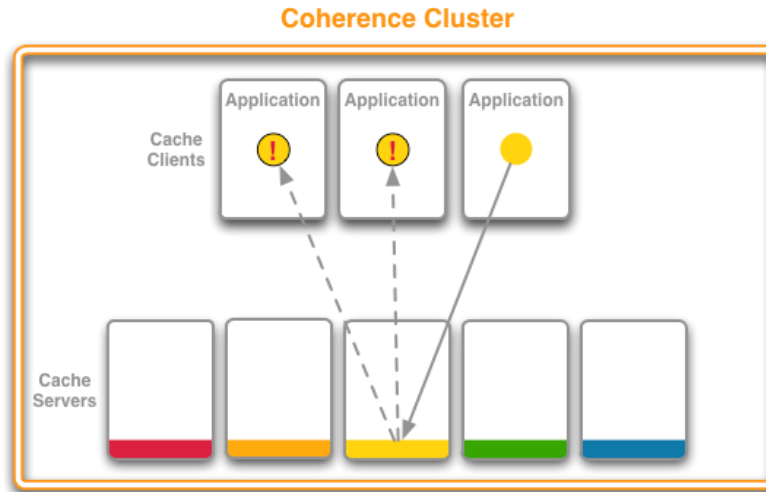


Coherence Cluster



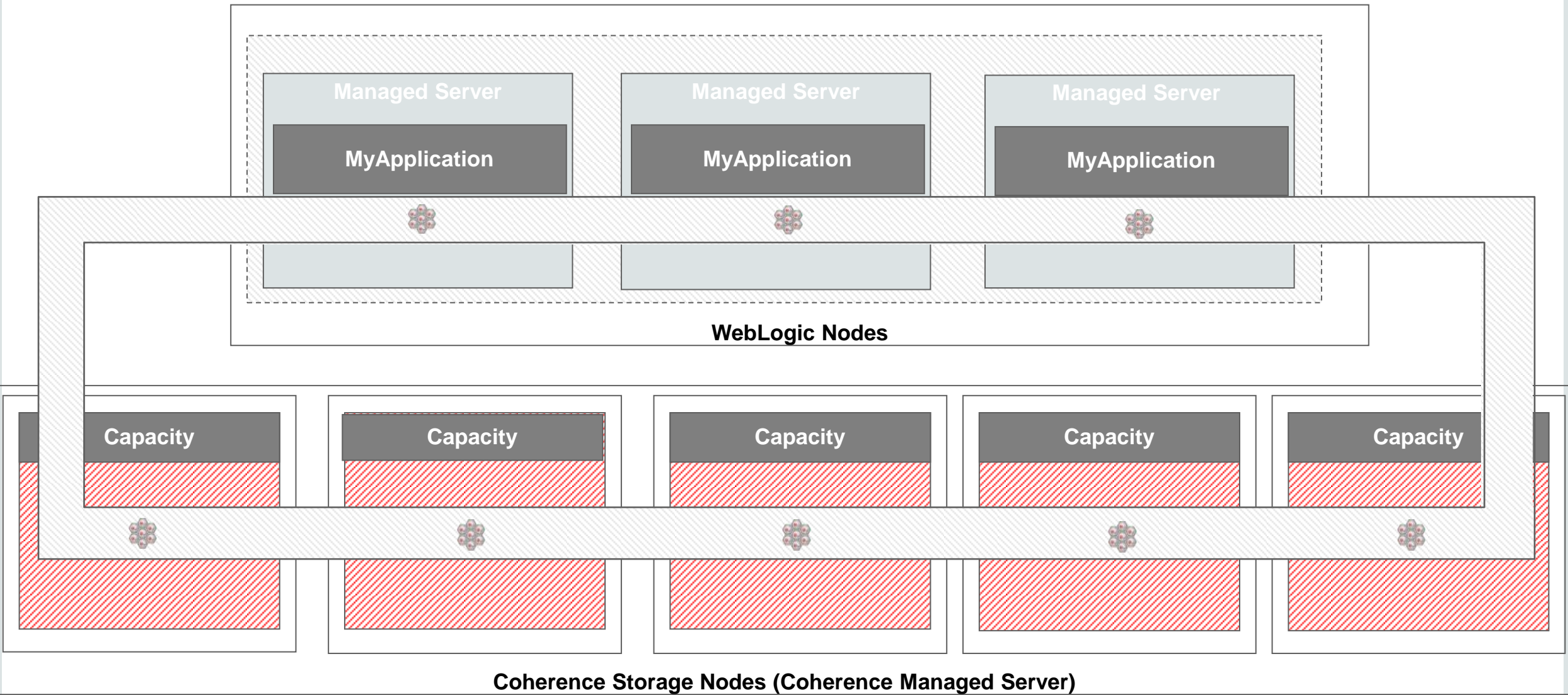
Parallele Verarbeitung: Filter mit Indizes, Aggregationen

# Coherence in a Nutshell



- Map Events: entryInserted, entryUpdated, entryDeleted
- Live Events: Partitioned Cache Events, Partitioned Service Events, Lifecycle Events

# Java EE UND In-Memory Grid



# Einige kritische Erfolgsfaktoren - Betriebssicht

- Netzwerkkonfiguration und -planung
  - Bandbreite und Latenz: Coherence Netzwerk Tools kommen mit!
  - Konfiguration Netzwerkkarten (Linkaggregation unter Solaris (aktiv-aktiv vs. aktiv-passiv), z.B. IP Multipathing, DataLink Multipathing, etc. bei Solaris)
  - Konfiguration Switches
  - Rebalancing im Ausfall!
- Sorgfältige Kapazitätsplanung:
  - Anzahl der Knoten/JVMs: Kapazität, Planung für Fehlertoleranz
  - HW Planung: JVM Knoten per Rechenknoten (RAM, Prozessorkerne)
  - <http://www.oracle.com/technetwork/middleware/coherence/planning-coherence-deployment-1985929.pdf>



# Einige kritische Erfolgsfaktoren - Entwicklersicht

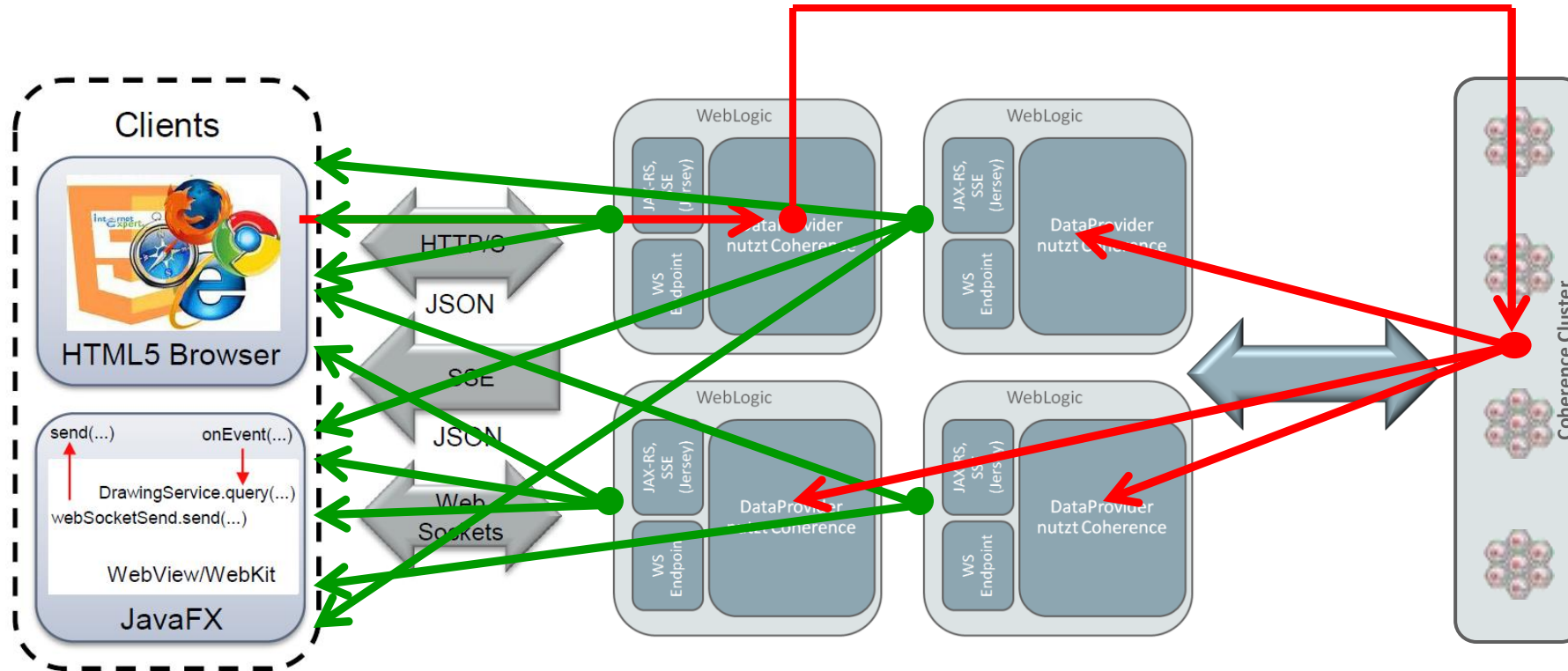
- Entscheide, was lokal oder verteilt benötigt wird
- Vermeide Standardserialisierungsmechanismus: implementiere POF (98:2 Regel)
- Bringe Code zu den Daten, nicht Daten zum Code
- Denke über Parallelisierung nach
- Überdenke Dein Objektmodell
- Suche Einfachheit und Eleganz: kein „Overengineering“ von Anforderungen:
  - Bitte keine RDBMS nachimplementieren!

# Hilfsmittel, Testen & Co

The screenshot shows a web browser window displaying the Oracle Coherence Community page on GitHub. The browser's address bar shows the URL `https://github.com/coherence-community`. The page header includes the GitHub logo, a search bar with the text "Search or type a command", and navigation links for "Explore", "Features", "Enterprise", and "Blog". A green "Sign up" button is also visible. The main content area features the Oracle Coherence Community logo and name. Below this, there is a search bar with the text "Find a repository..." and a "Filters" dropdown. Two repositories are listed: "oracle-tools" (Java, 9 stars, 8 forks) and "coherence-incubator" (Java, 24 stars, 24 forks). A "Members" sidebar on the right lists "aseovic" and "brianoliver" (Brian Oliver).

# Demos

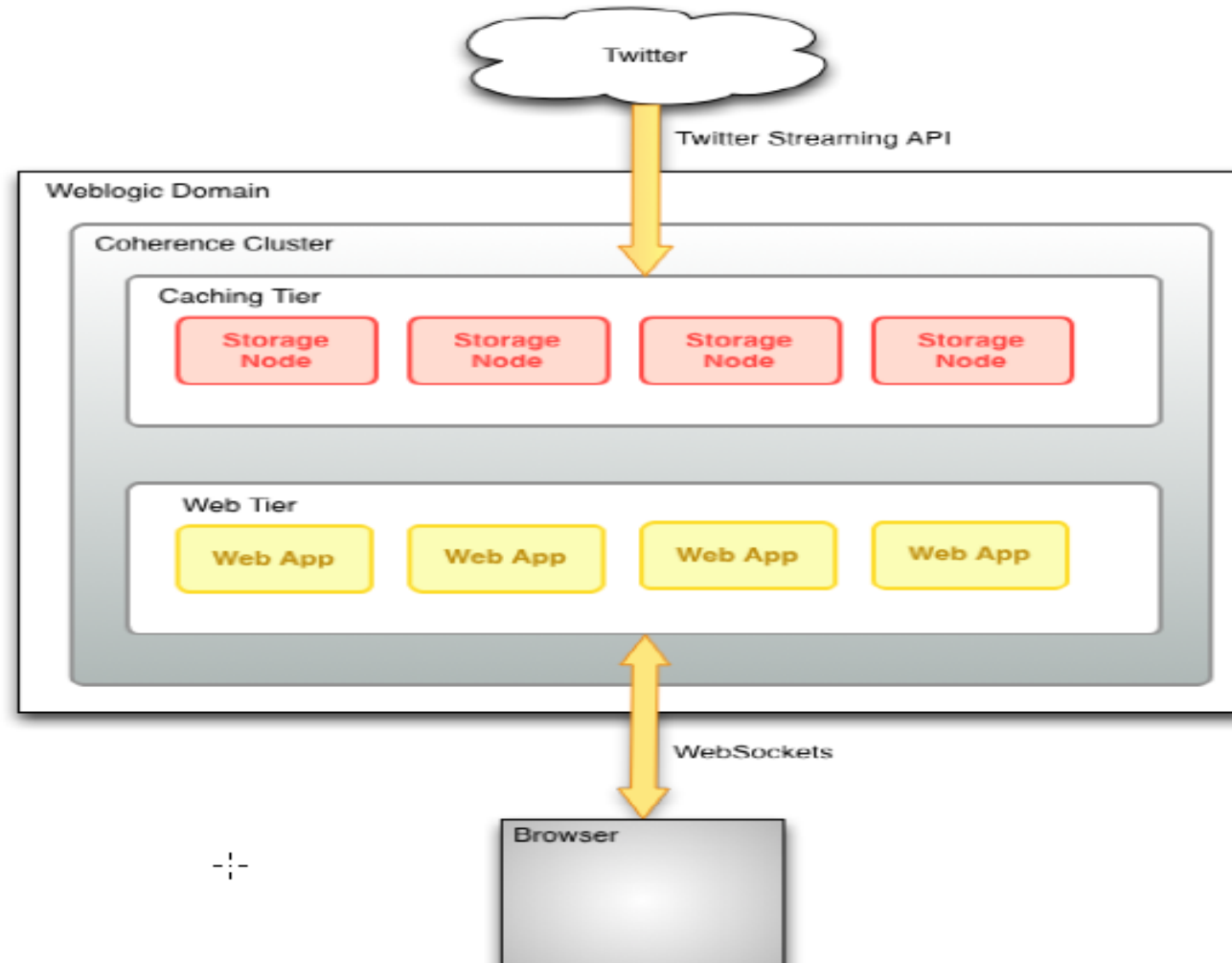
# Skalierbare Architektur: Eventbasiert



<https://github.com/doschkinow/hol-sse-websocket>, solutions/exercise9

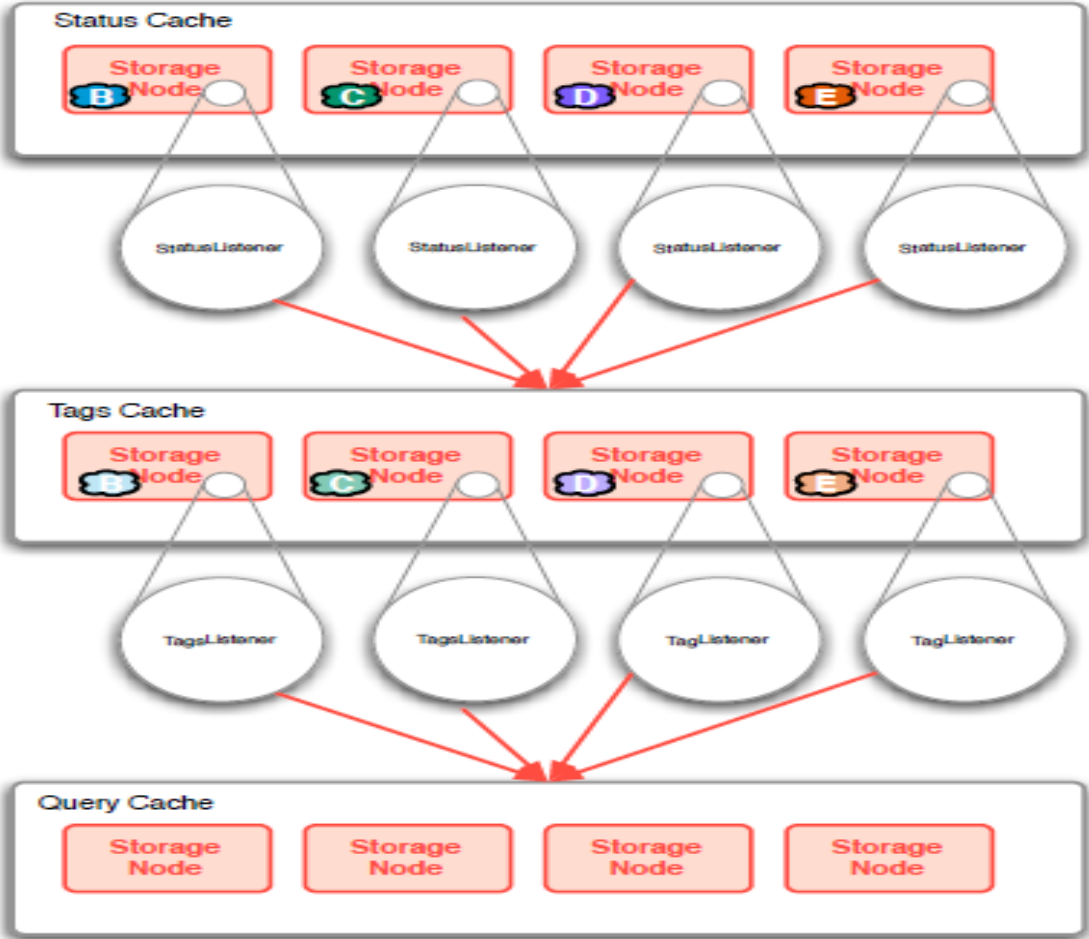
# Demo 1/3

## Tweeter Streaming API with a Coherence and WLS Cluster using Avatar



# Demo 2/3

## The Caching Tier





# Zusammenfassung

- P2P Clustering and Data Management Technology
- No Single Points of Failure
- No Single Points of Bottleneck
- No Masters / Slaves / Registries etc
- Scale to limit of the back-plane
- Use with commodity infrastructure
- Linearly Scalable By Design

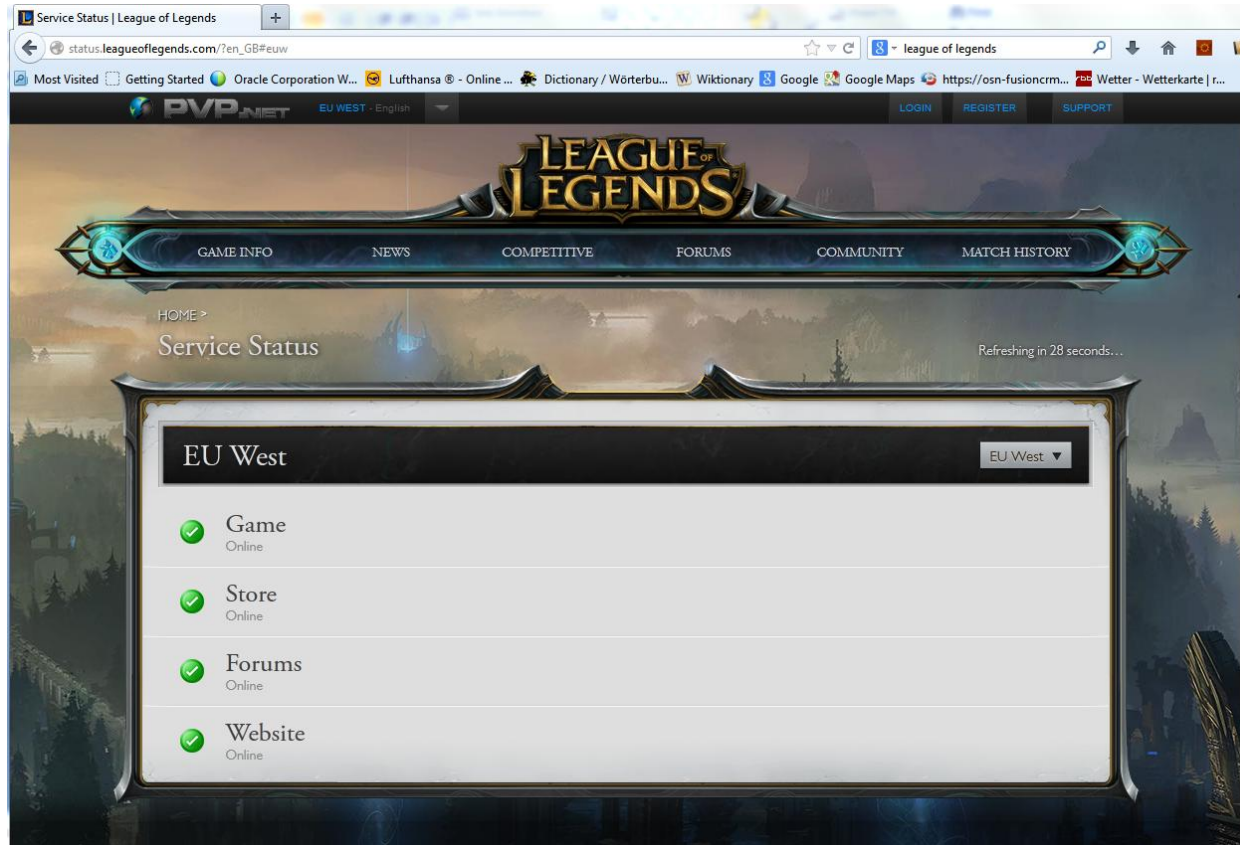
**ORACLE**  
Coherence

**ORACLE**  
FUSION MIDDLEWARE  
WEBLOGIC SERVER

Aber auch mit , , ... 😊 . Try it.



# Real-Life Use Case: Online Game League of Legends



<http://www.infoq.com/presentations/League-of-Legends>

[http://qconsf.com/sf2010/dl/qcon-sanfran-2010/slides/RandyStafford and ScottDelap LeagueOfLegendsScalingToMillionsOfNinjasYordlesAndWizards.pdf](http://qconsf.com/sf2010/dl/qcon-sanfran-2010/slides/RandyStafford%20and%20ScottDelap%20LeagueOfLegendsScalingToMillionsOfNinjasYordlesAndWizards.pdf)

<http://www.oracle.com/technetwork/middleware/coherence/coherence-case-studies-091909.html>

# Join the Coherence Community



@OracleCoherence



/OracleCoherence



/OracleCoherence



Oracle Coherence  
Users



blogs.oracle.com/  
OracleCoherence

coherence.oracle.com

# Java Application Server Plattform Community

Eine Community von ORACLE für Kunden, Partner und Interessierte

Code Camps

Demos

WebLogic Server

GlassFish Server

Community Treffen

Java EE

Vorträge

JSRs

Serverseitige Entwicklung

Administration

Wissensaustausch



Registrierung:

<https://www.xing.com/net/oraclejavaappserver>

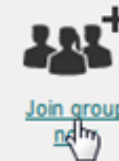
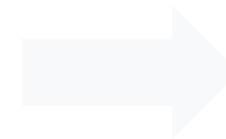
Blog:

<http://fmwtech.wordpress.com>

Ansprechpartner:

[michael.braeuer@oracle.com](mailto:michael.braeuer@oracle.com)

[peter.doschkinow@oracle.com](mailto:peter.doschkinow@oracle.com)



[Join group](#)



[Bookmark this group](#)

**ORACLE®**