



WRITE ONCE.  
SCALE ANYWHERE.

## The End Of Application Servers as we know them



Nati Shalom

CTO GigaSpaces Technologies

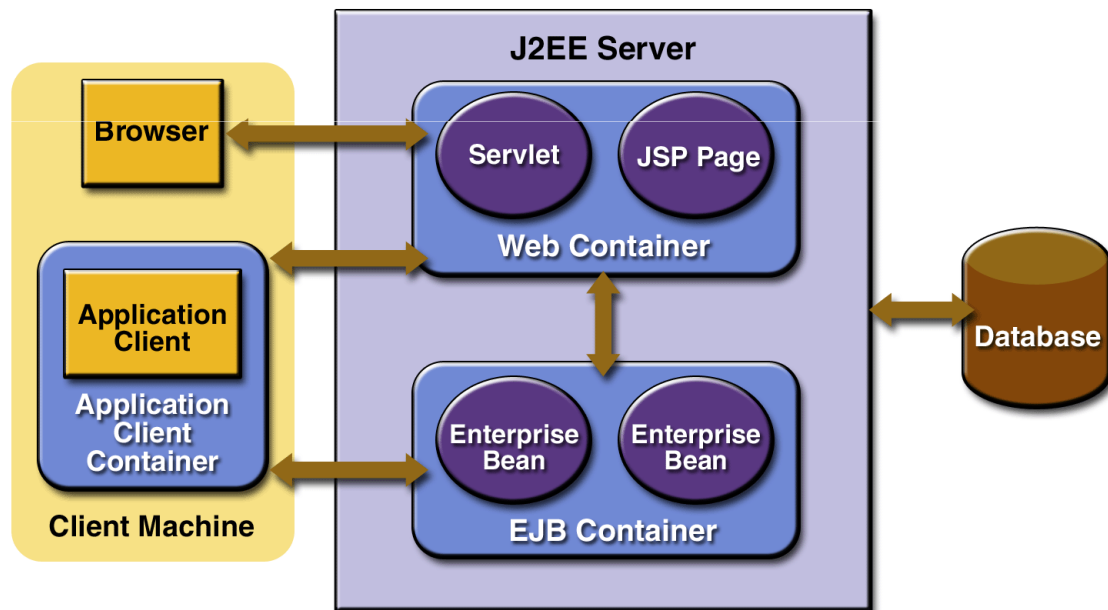
[natishalom.typepad.com](mailto:natishalom.typepad.com)

# Application Server Defined (Not just JEE)

- Web Container
- Business Logic Container (EJB container in the J2EE world)
- Acts as a database front-end
- Clustering
- High-Availability
- Resource Management: Pooling of database connections, threads, other
- Transactions - Coordination
- Security
- JMS – Messaging, Service Coordination

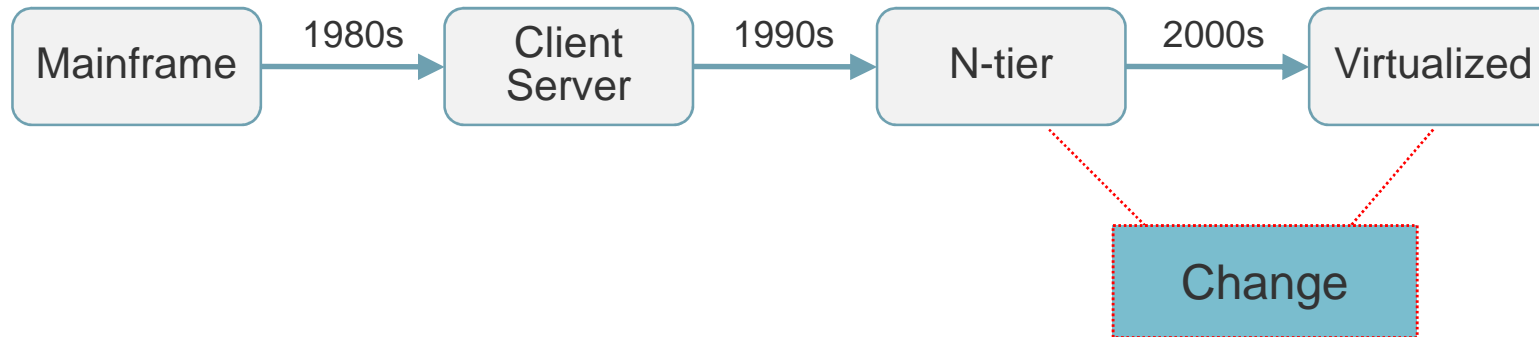
# JEE Application Server

- Java centric
- Database centric
- Tier based
- A bunch of standard API (JSP, Servlets, EJB, JPA, JDBC, JMS,..)



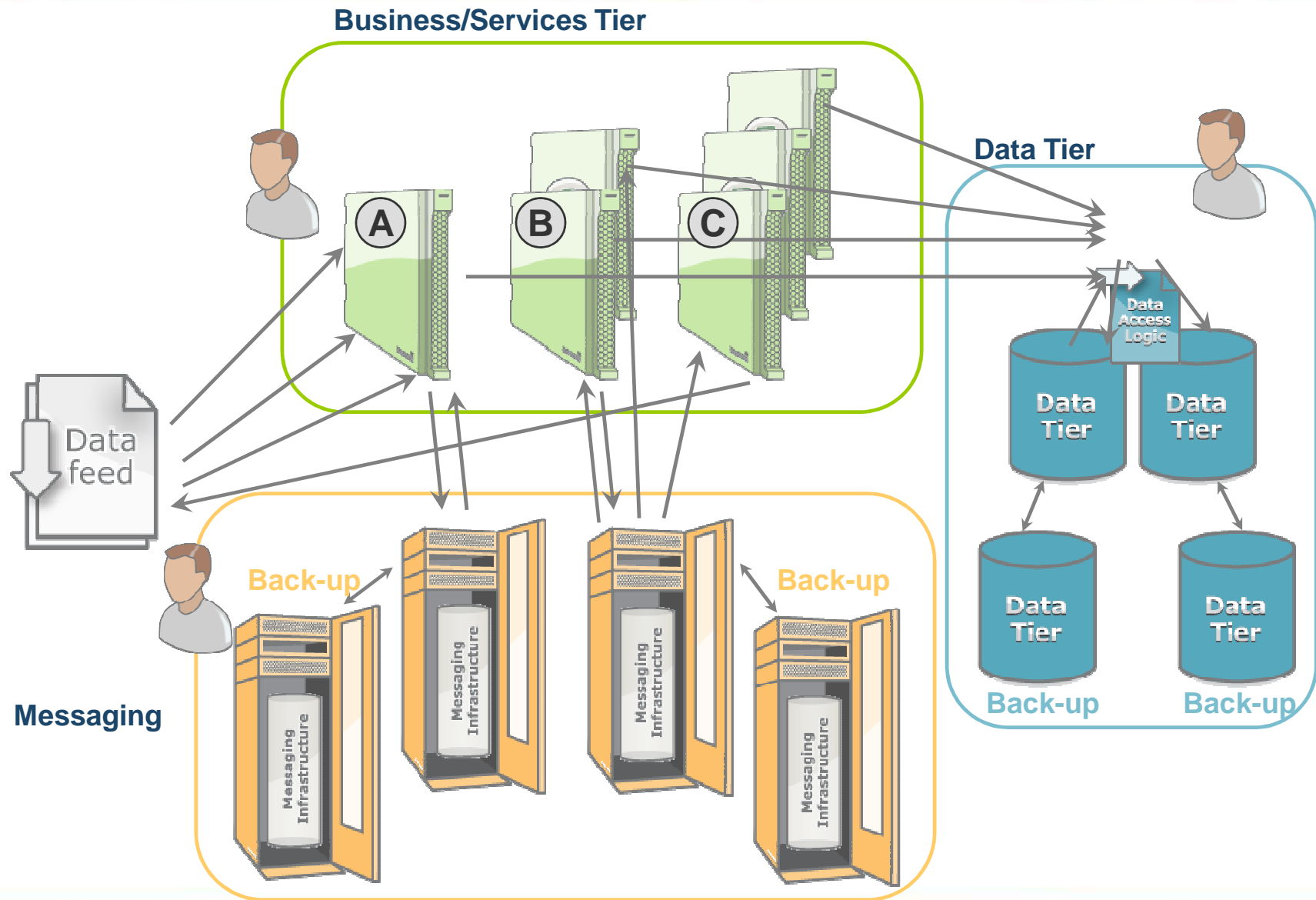
# Market Trends

- Software architecture is undergoing a “once in a decade” transformation:



- Each paradigm transformation undermines the existing technologies. The current transformation signals the decline of traditional application-servers, RDBMSs etc.
- The new paradigm is virtualization, grid, cloud computing, PaaS, SaaS
- IDC on Cloud Computing: "This is about the IT industry's new model for the next 20 years,"
  - Vernon Turner, head of enterprise infrastructure, consumer and telecoms research.

# Moving from N-Tier to Virtualized: Do You See the Problem?



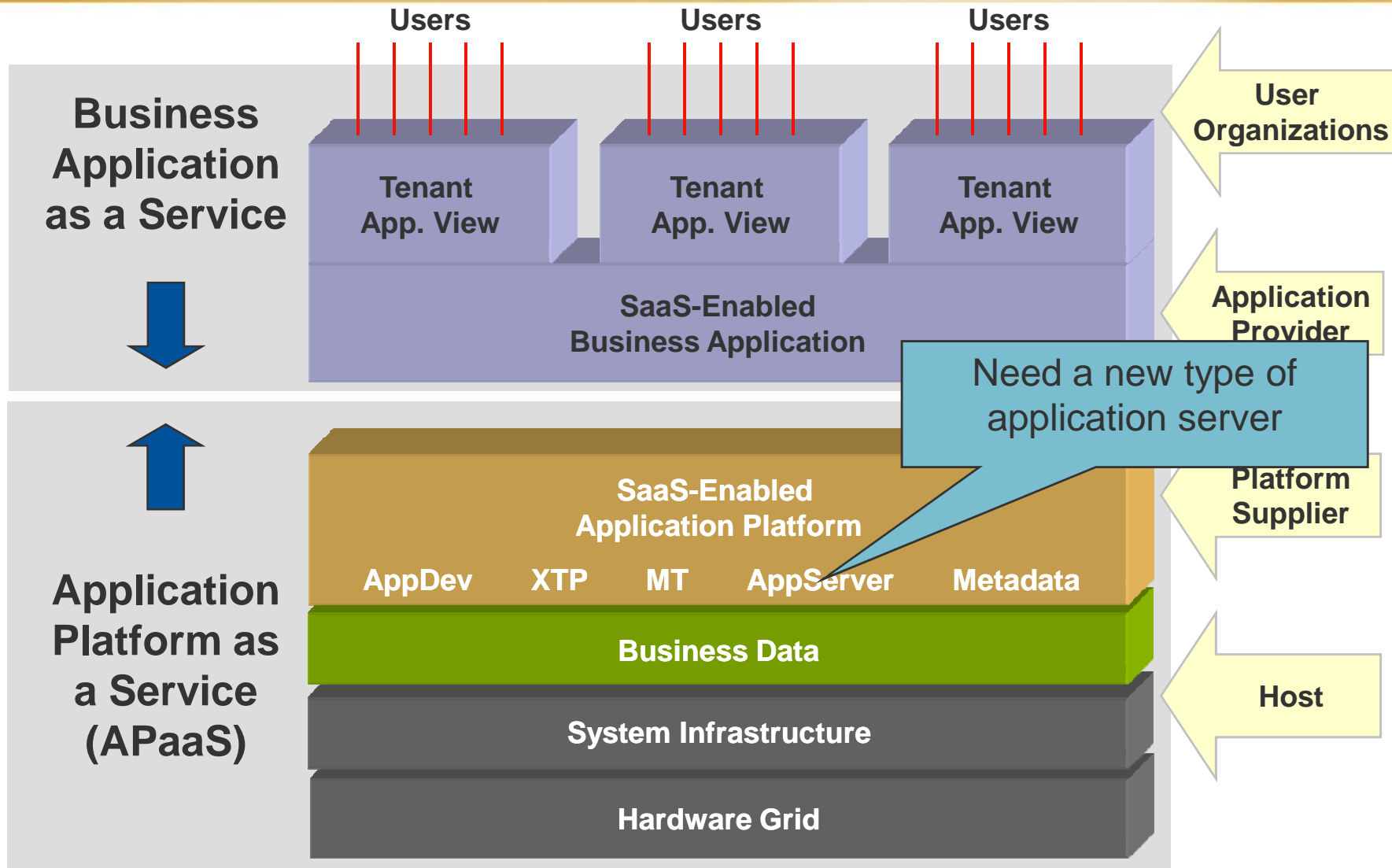
# The need for change



# The need for change

- Support for new architecture (Event Driven, Grid, Cloud, Multi Core,..)
- Not just Java (support .Net, ..)
- Support wider variety of applications (Telco, RT analytics, Batch, Parallel)
- Database – less centric , support for In Memory Data Grids
- Scalable Transaction model – Alternative to 2pc/XA
- Support for Stream based programming

# The future: Application Platform as a Service (Gartner)



XTP — Extreme Transaction Processing    MT — Multitenancy



# The current market place

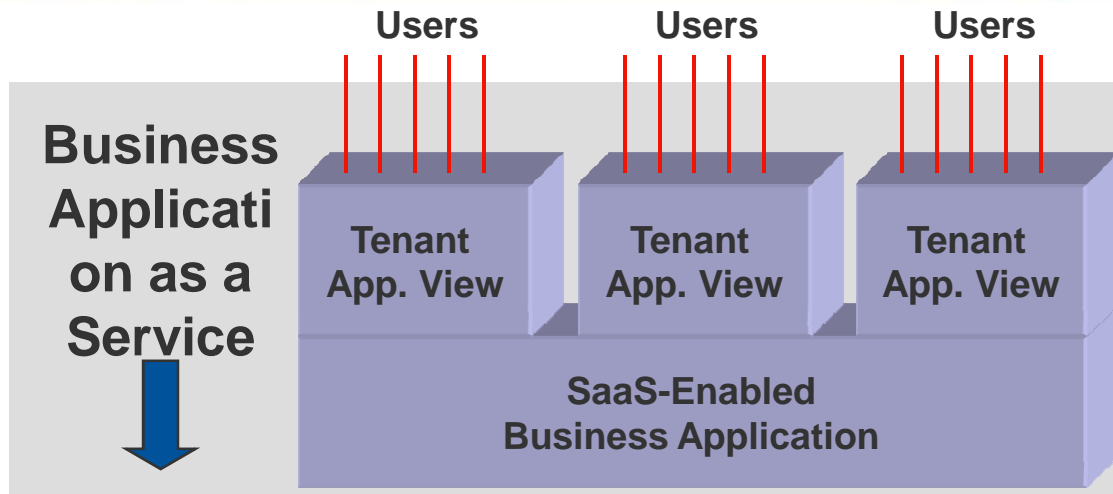
The change is already underway

- JEE Players (More of the same)
  - Oracle – Scale Out application server (WL+Coherence+...)
  - IBM – Web Sphere XD + Extreme Scale
  - JBoss - ???
- New players
  - GigaSpaces – scale out app server
  - Spring Source - dynamic containers
- New niche players
  - GridGain, - Adding simple grid to Java/JEE
  - Terracotta – Clustered JVM
- Non JEE Players (PaaS Players)
  - Microsoft Azure
  - Google App Engine
  - Sales Force – Force.com

# Which way to go?

- Avoid radical change
  - Look for a solution that will enable gradual change
- Choose architecture that fits the scale-out model
- Avoid vendor lock-in
  - Look for solution that could run on more than one product
  - Minimize lock in:
    - Use of standard where possible (be aware of lack of standards)
    - Abstraction – decouple the application from the implementation
- Future proof your application
  - Don't make decision today, but be ready to make one without major effort
  - Avoid long-term commitment – choose the right licensing model

# How? Middleware virtualization

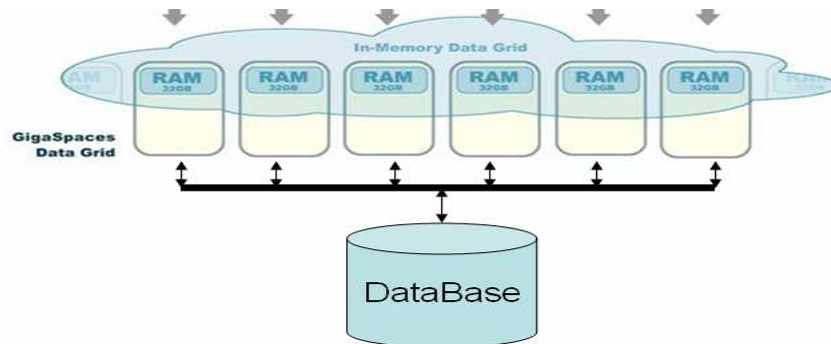


## Middleware Virtualization



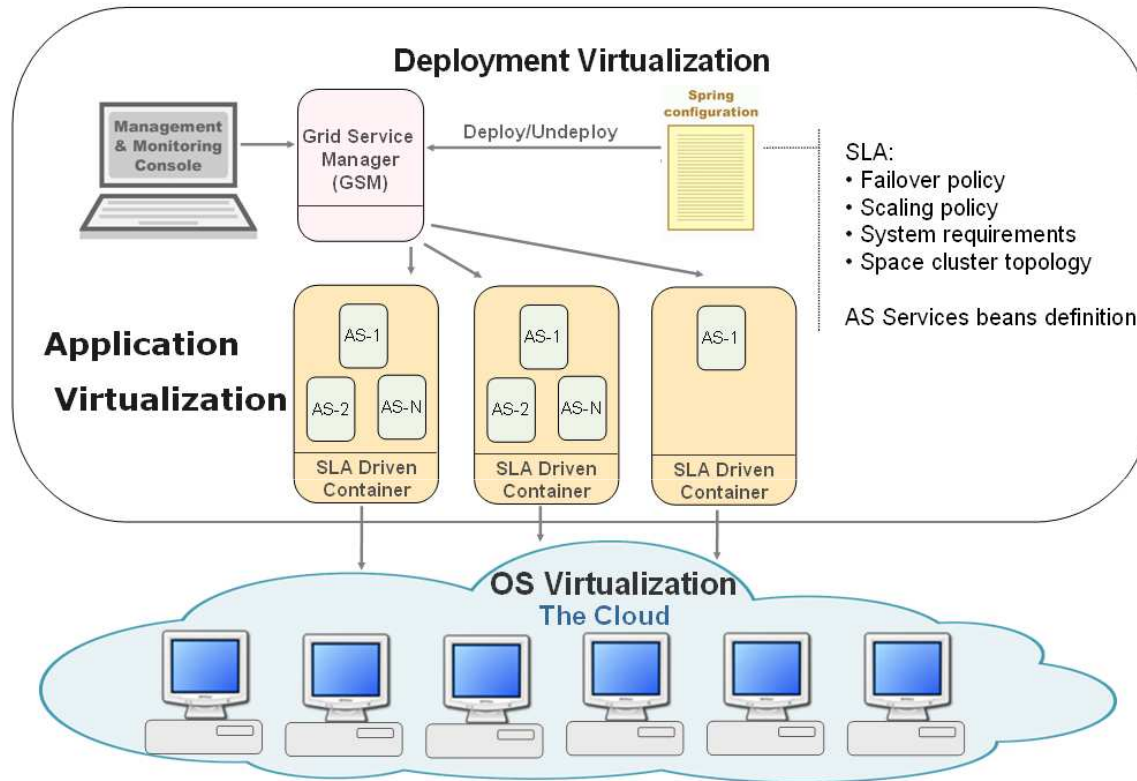
### OpenSpaces – GigaSpaces Development framework

Messaging Services Data



- "All problems in computer science can be solved by another level of indirection" ([Butler Lampson](#))
- Similar principles to storage virtualization
- Decouple the application from the deployment environment
- Use partitioning to split the load and the data.
- Support JEE, .Net

# Automate manual process



- Move manual process to SLA driven deployment
- Scale only when you need
- Self healing

# Plugging virtualized middleware in seamless manner

- Web
  - Deploy standard WAR on web processing unit
  - Session replication can be enabled through configuration
  - Choose web container (Jetty, Glassfish, Tomcat,..)
- Messaging / event processing
  - MDBs can be easily replaced with other event listeners
- Remoting
  - Export your service using new remoting proxy
- Data access
  - Use 2<sup>nd</sup> cache for Hibernate (Very limited approach)
  - Convert your DAOs to DataGrid, use async persist
- TX management
  - Use Spring declarative transactions...

# Converting Your Messaging Layer

```
@EventDriven
@Polling
@TransactionalEvent
public class OrderMatcher {
    private OrderDao orderDao;
    private TradeDao tradeDao;

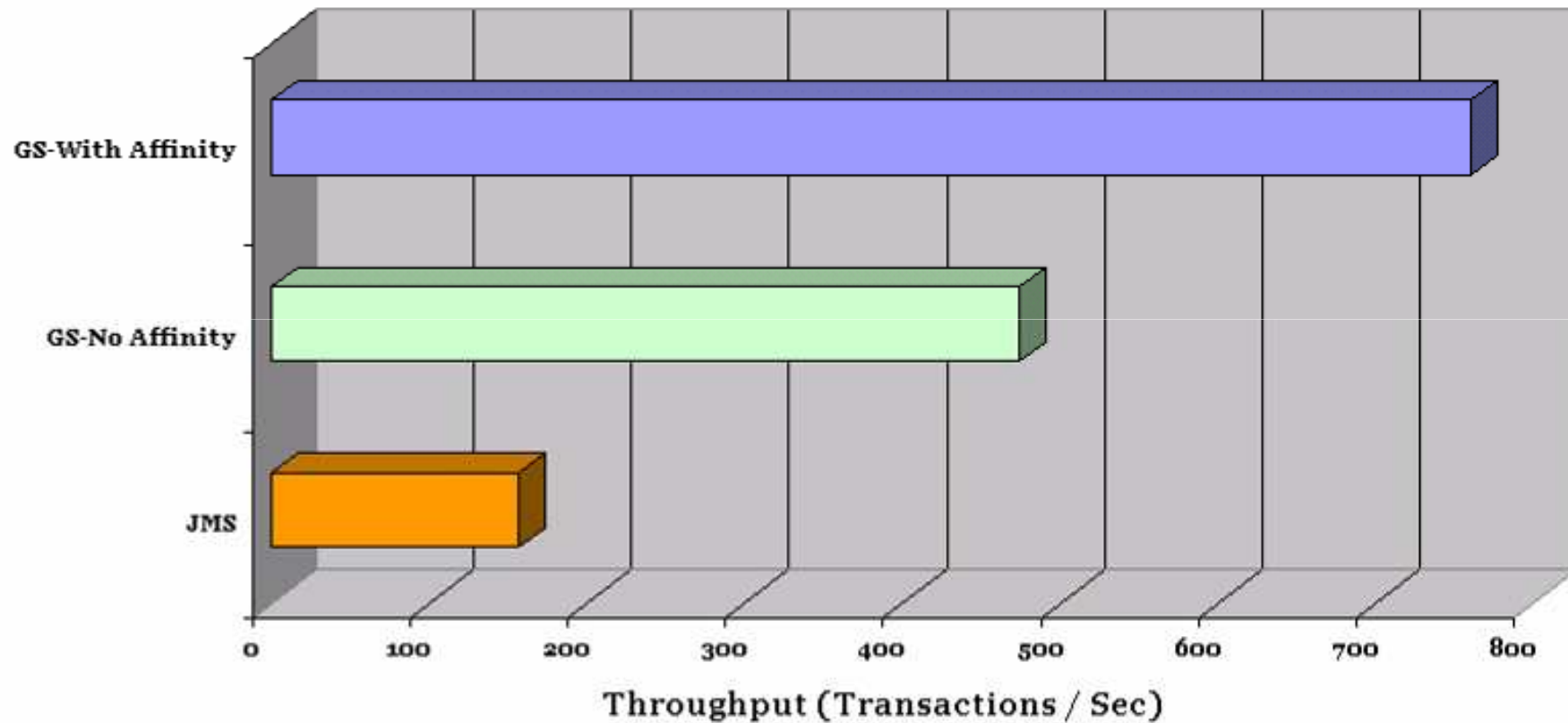
    public OrderMatcher() {}

    public OrderMatcher(OrderDao orderDao, TradeDao tradeDao) {
        this.orderDao = orderDao;
        this.tradeDao = tradeDao;
    }

    @EventTemplate
    public Data unmatchedAskOrder() {
        Order template = new Order();
        template.setStatus(Order.STATUS_ACCEPTED);
        template.setIsBid(false);
        return template;
    }
}
```

# Converting Your Messaging Layer

Throughput - JEE vs GigaSpaces  
(Higher Is Better)



# Converting Your Remoting Layer

- Usually implemented by SLSBs with JEE
- No or minor code changes needed
- You get:
  - Location transparency
  - Dynamic service discovery
  - Synchronous or asynchronous invocation
  - Cluster wide invocations (Map/Reduce)
  - Data affinity
  - Automatic failover



## Dias nummer 16

---

**MSOffice1** I would write No / Minor code changes needed.  
The "No or minor" does not read well, I think  
; 26-10-2008

# Converting Your Remoting Layer

- Service Interface

```
public interface IBusinessService {  
    Result processInput(@Routing("getType") Input input);  
}
```

---

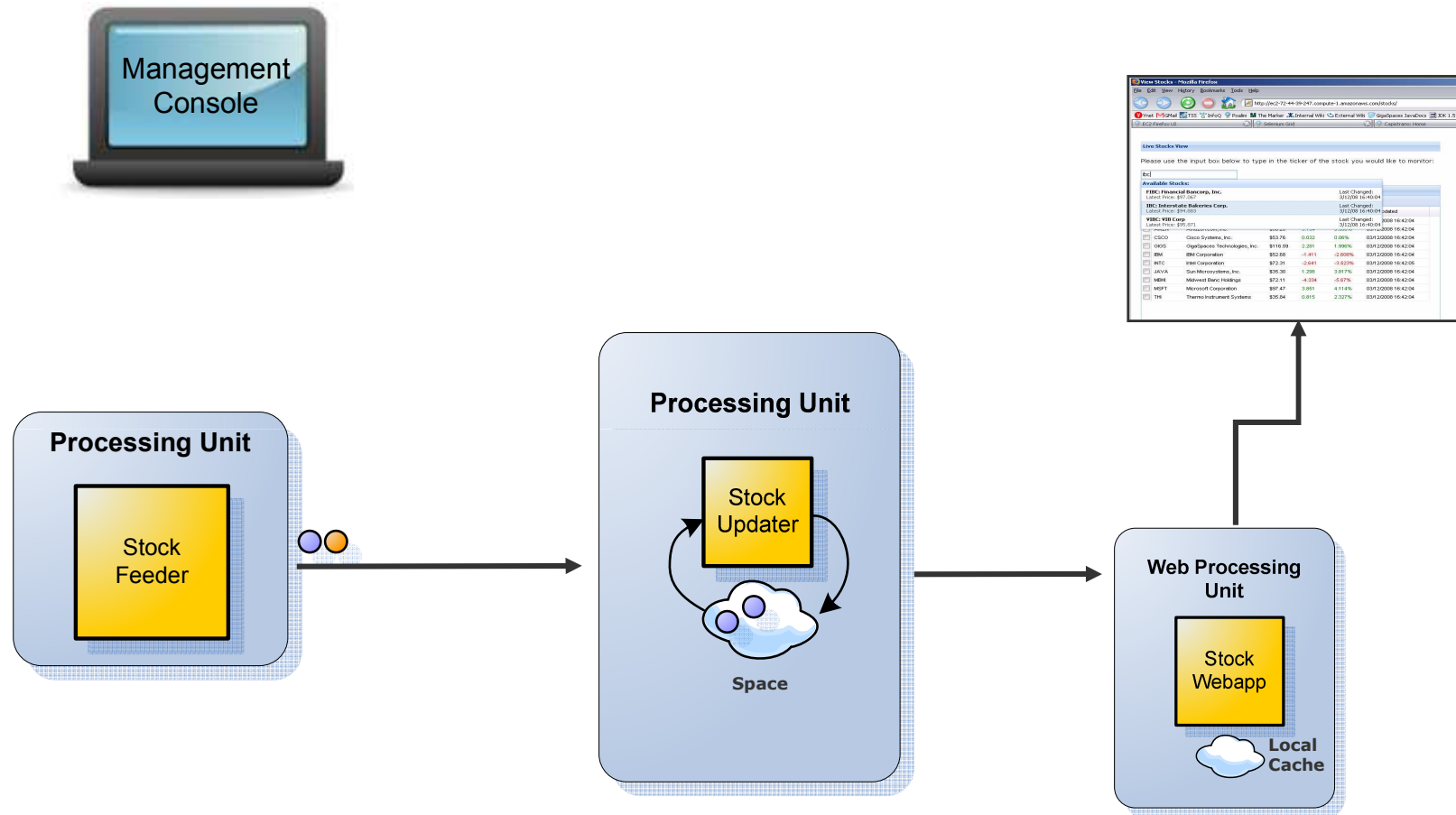
- `@ExecutorProxy //@SyncProxy, @AsyncProxy also possible`

- `IBusinessService dataProcessor;`  
`<os-remoting:annotation-support />`
- 

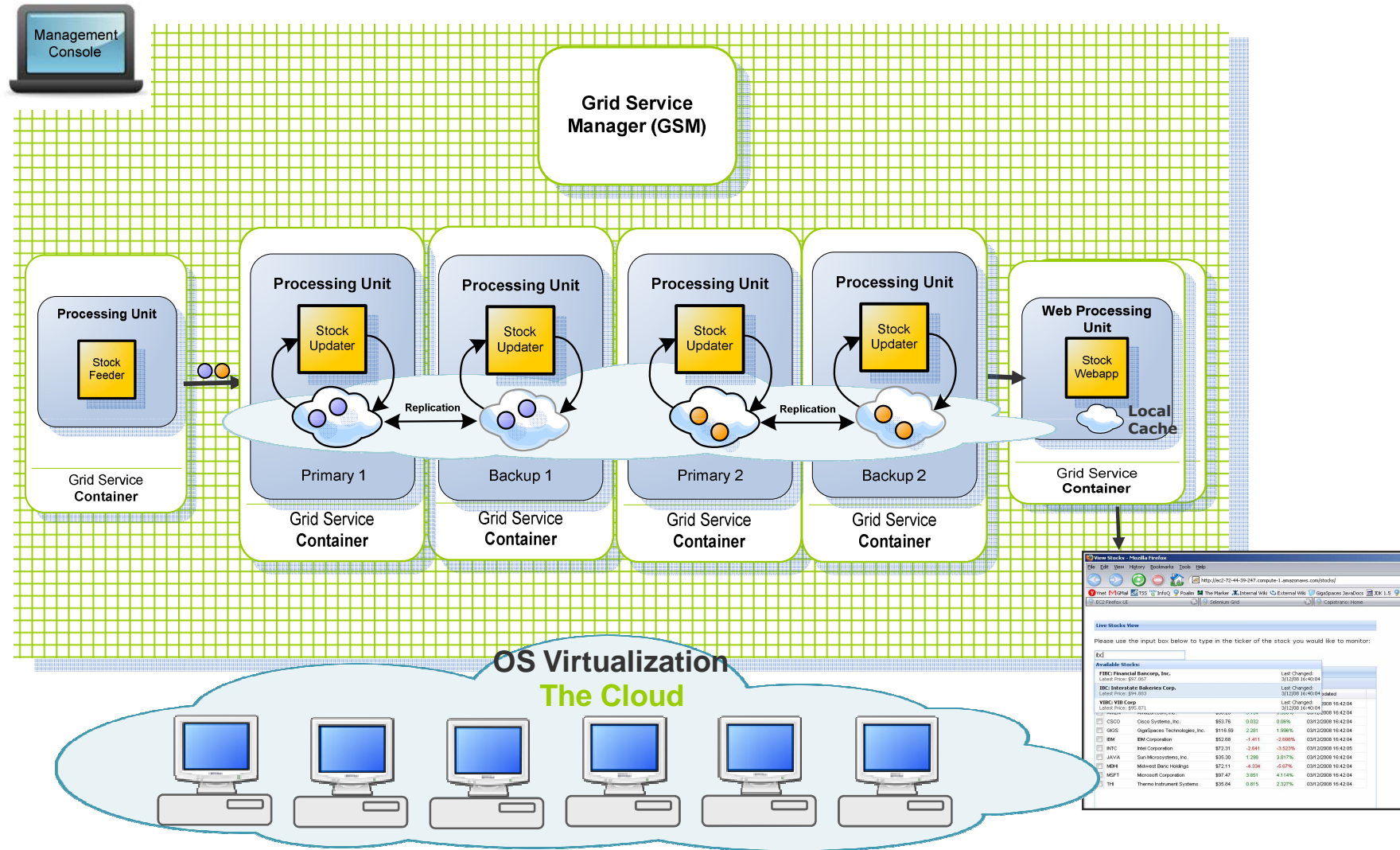
```
@RemotingService  
public class BusinessServiceImpl implements  
    Result processInput(Input input);  
        //do something meaningful  
    }  
}  
<context:component-scan base-package="com.mycompany.myproj" />  
<context:annotation-config />  
<os-remoting:annotation-support />
```

# Demo

# Demo Application – Logical Architecture



# Demo Application – Physical Architecture

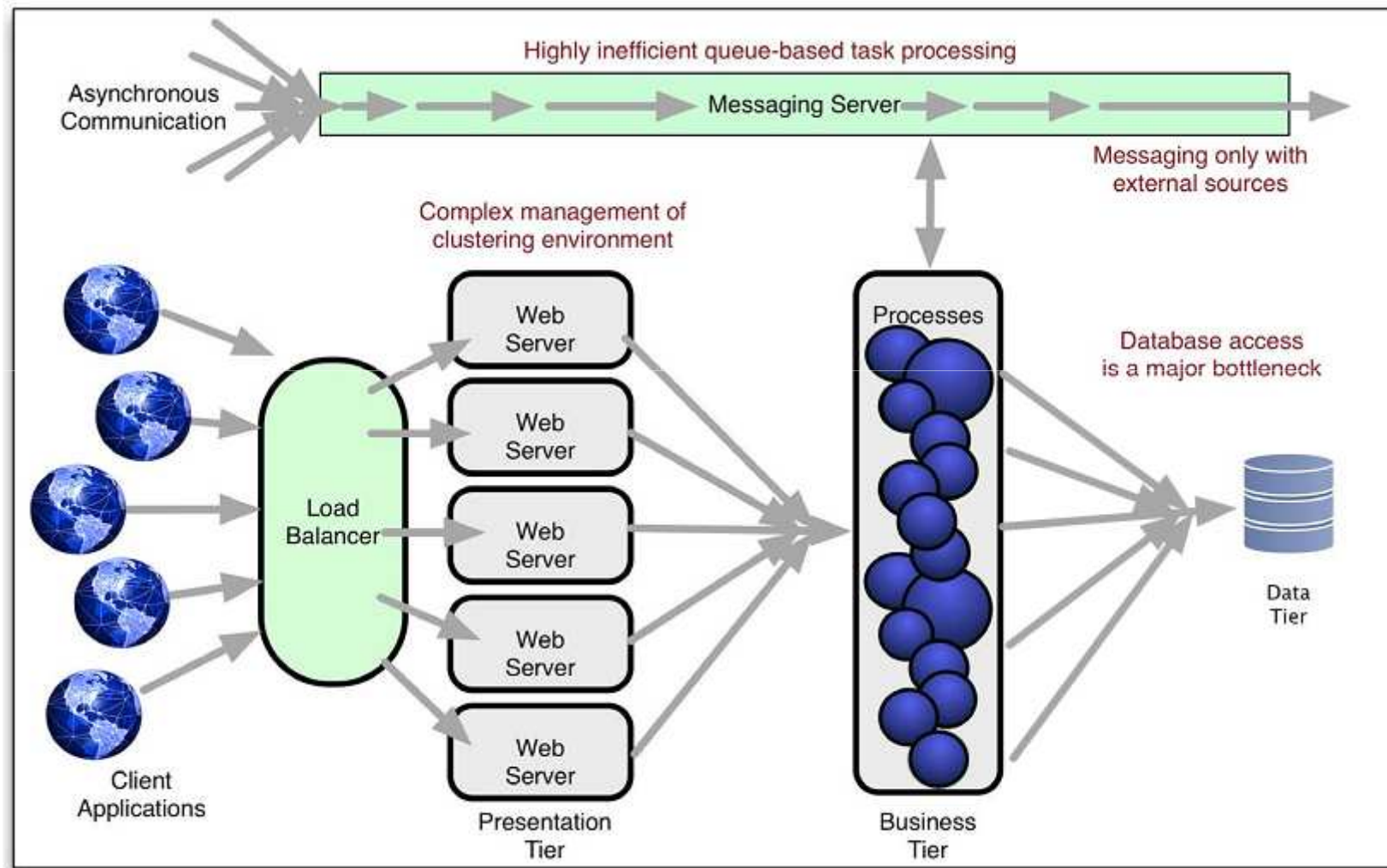




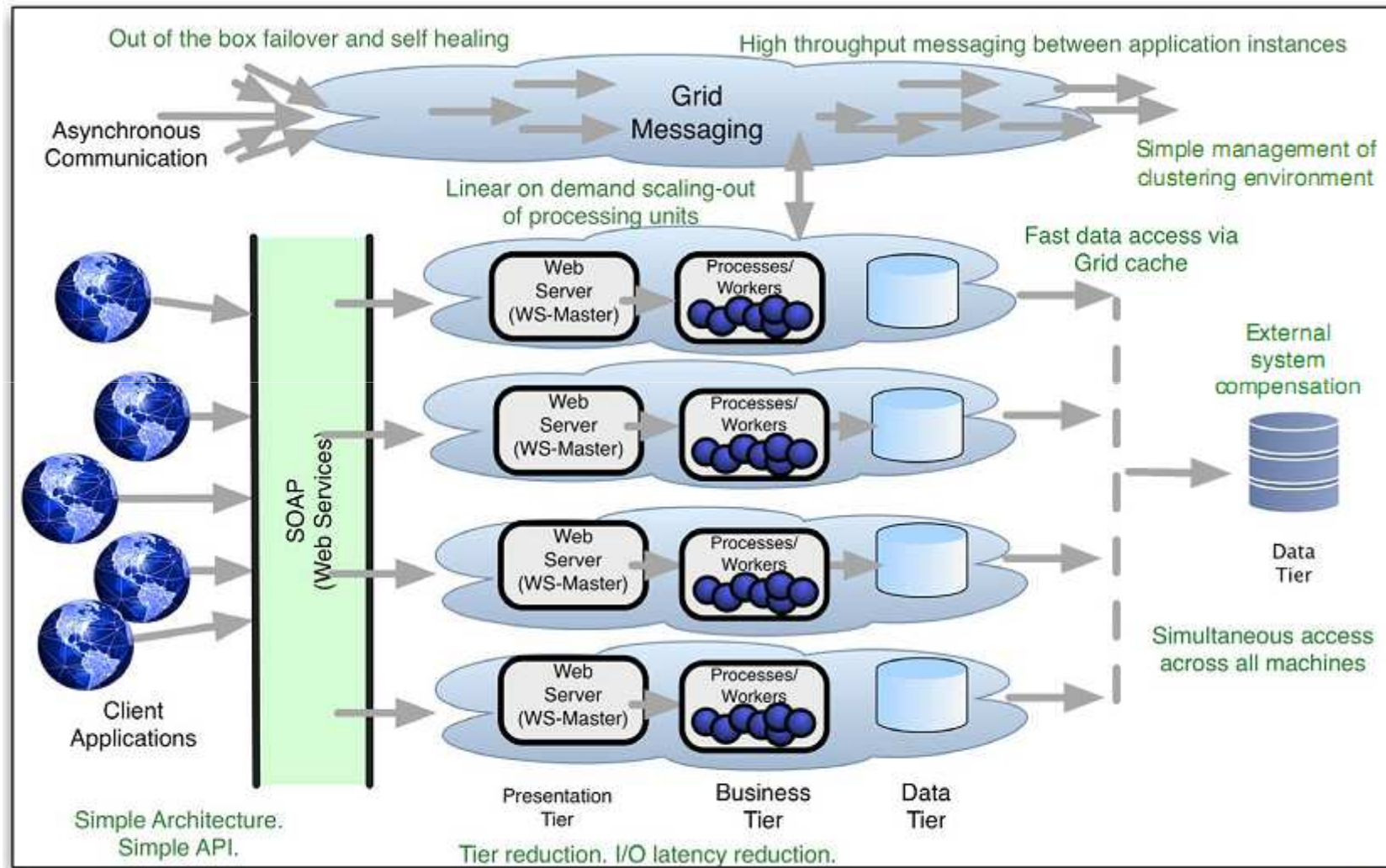
## Real Life Case Study



# Online OMS: Old System



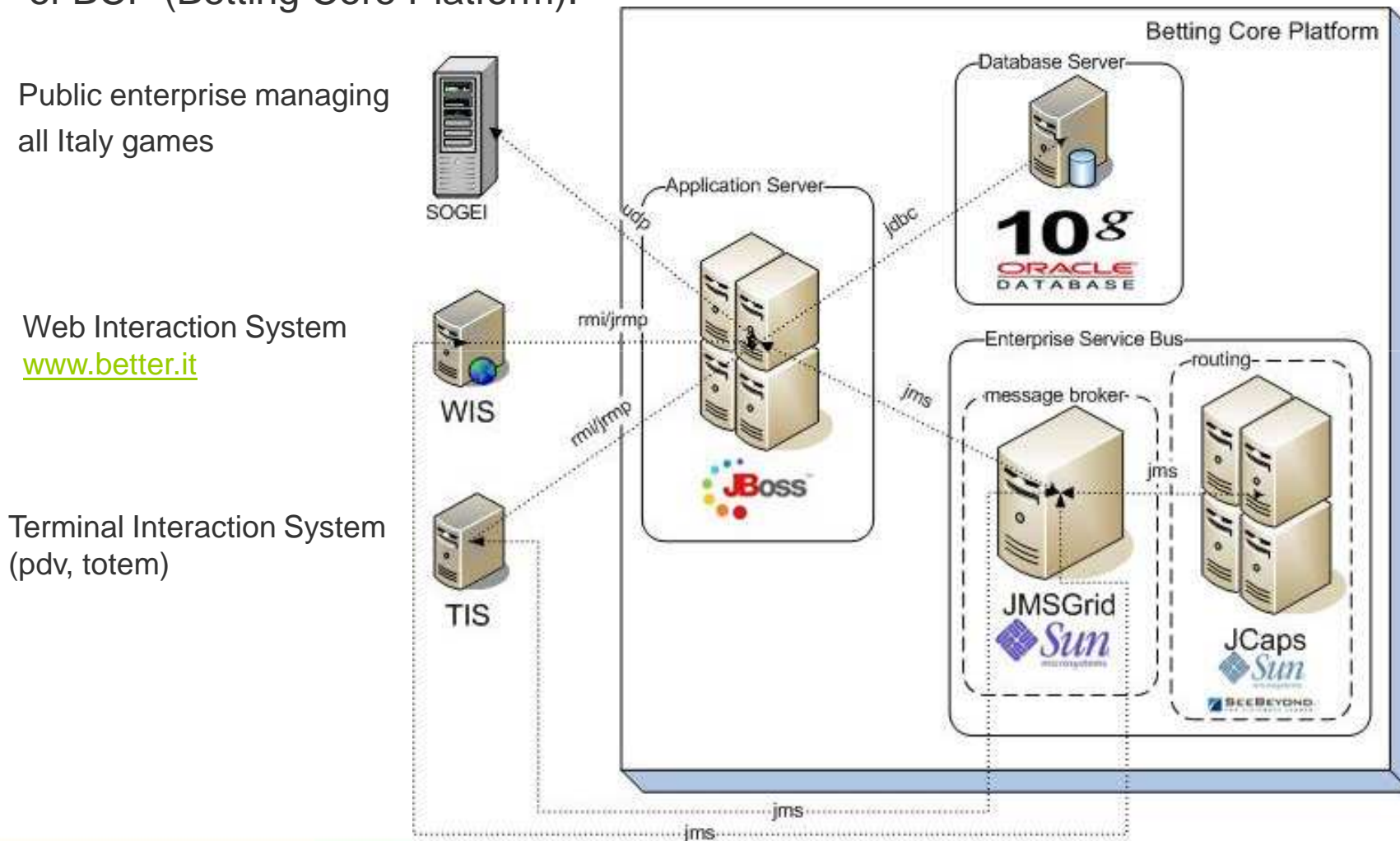
# Online OMS: SBA Enabled



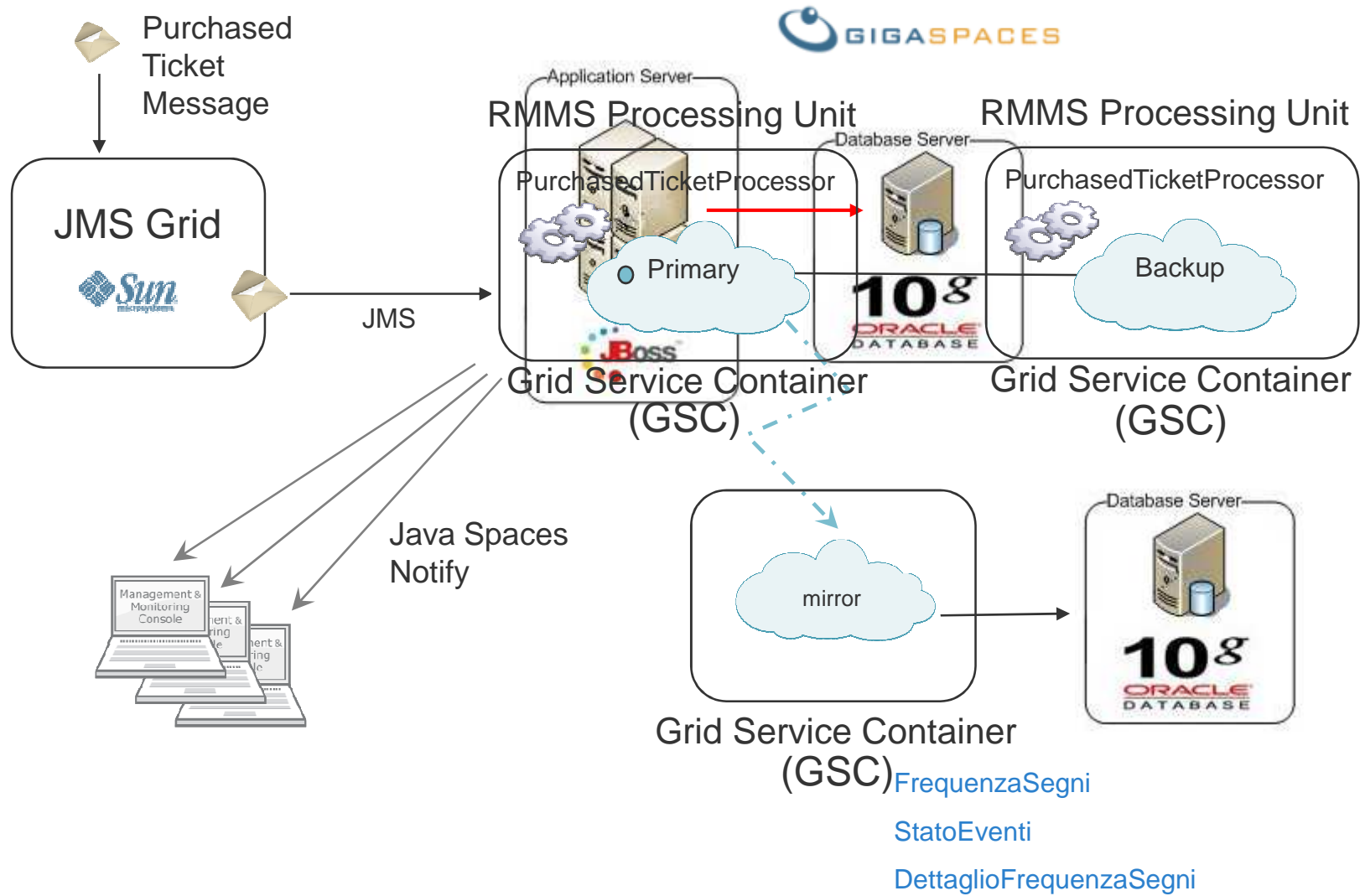


# Online Gaming: Old System

The following figure depicts an overview of the global application architecture of BCP (Betting Core Platform).

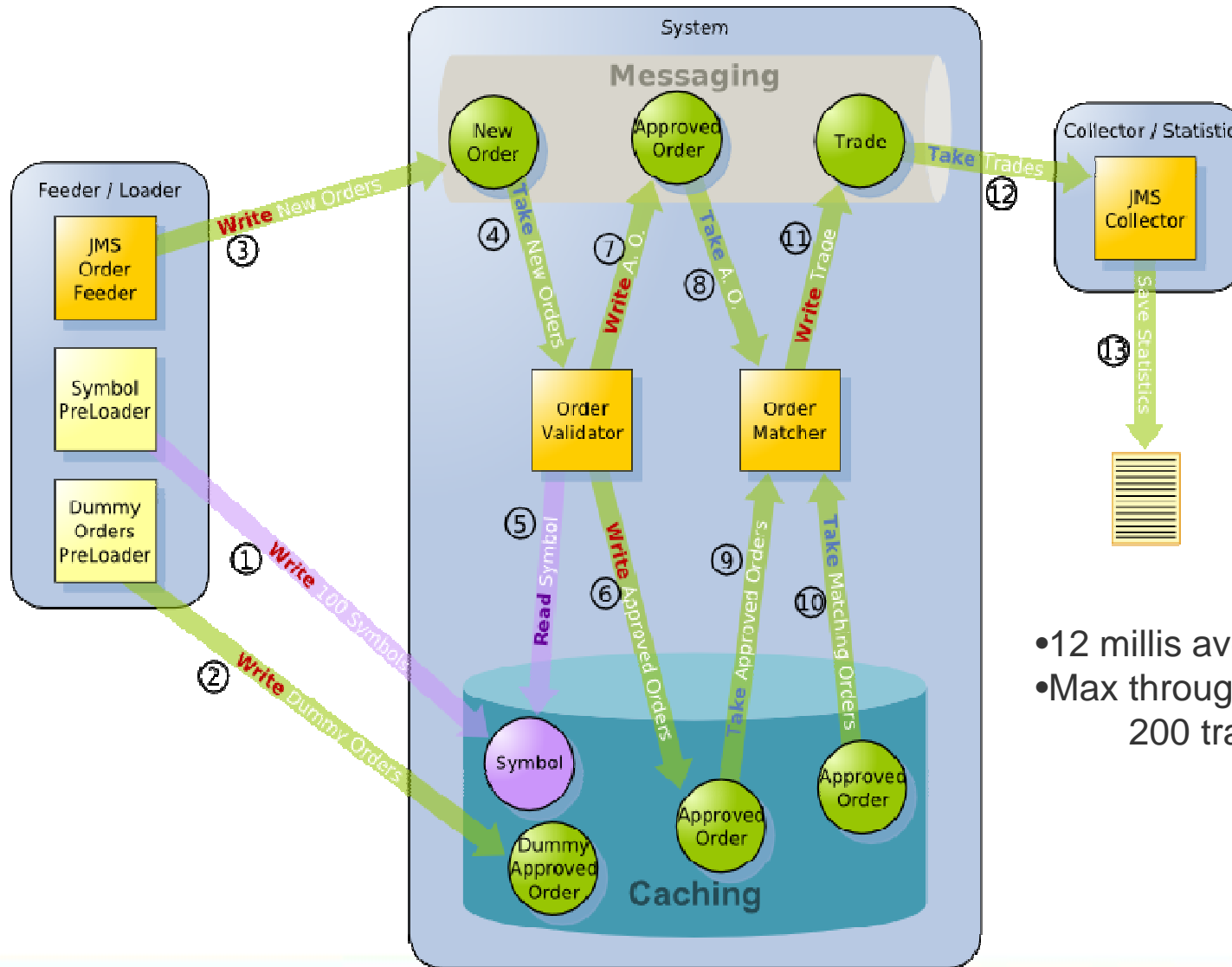


# Online Gaming: SBA Enabled



# Side By Side Comparison *Caching*

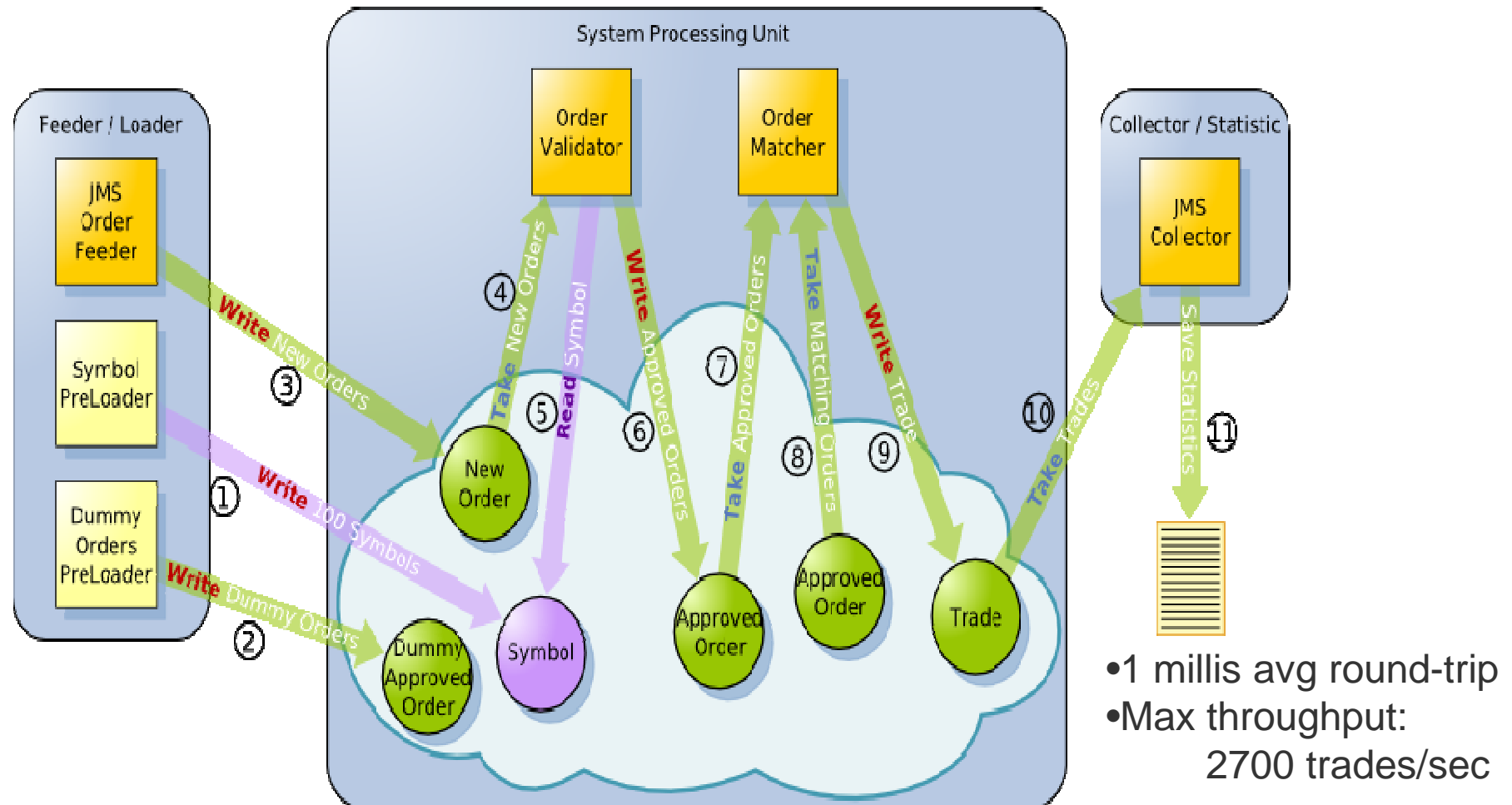
Space Based Architecture versus Tiers Based Architecture: TDA Workflow



- 12 millis avg round-trip
- Max throughput:  
200 trades/sec

# Side By Side Comparison SBA

Space Based Architecture versus Tiers Based Architecture: SBA WorkFlow



# Summary

- Change is unavoidable cope with it
- There are different way to cope with the change while minimizing the risk
- The key is to establish a clear road-map and avoid making decision that will lock you to a specific solution.

**Thank YOU**