

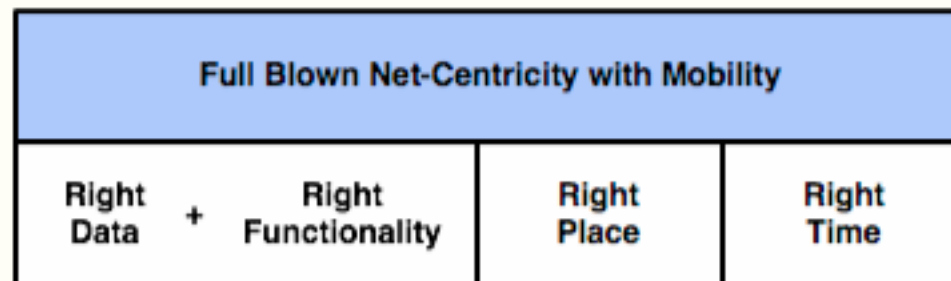
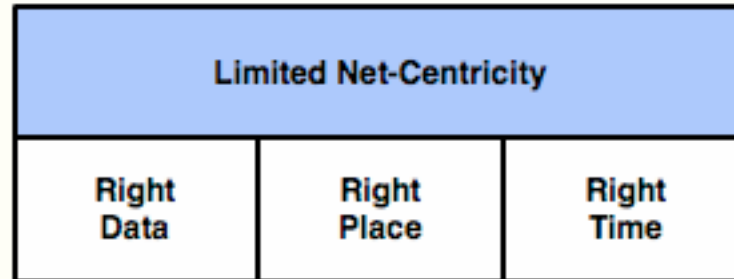
The Use of Mobile Object Technology in a Net-Centric System

May 5, 2008

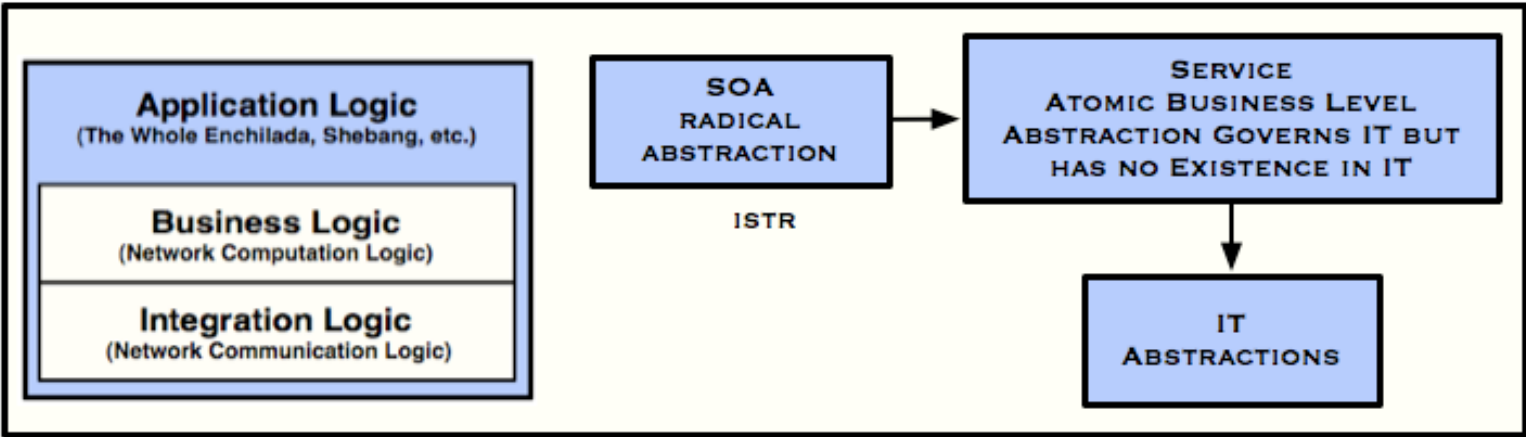
Michael McGrady
Senior Engineer
Topia Technology, Inc.

Preliminaries

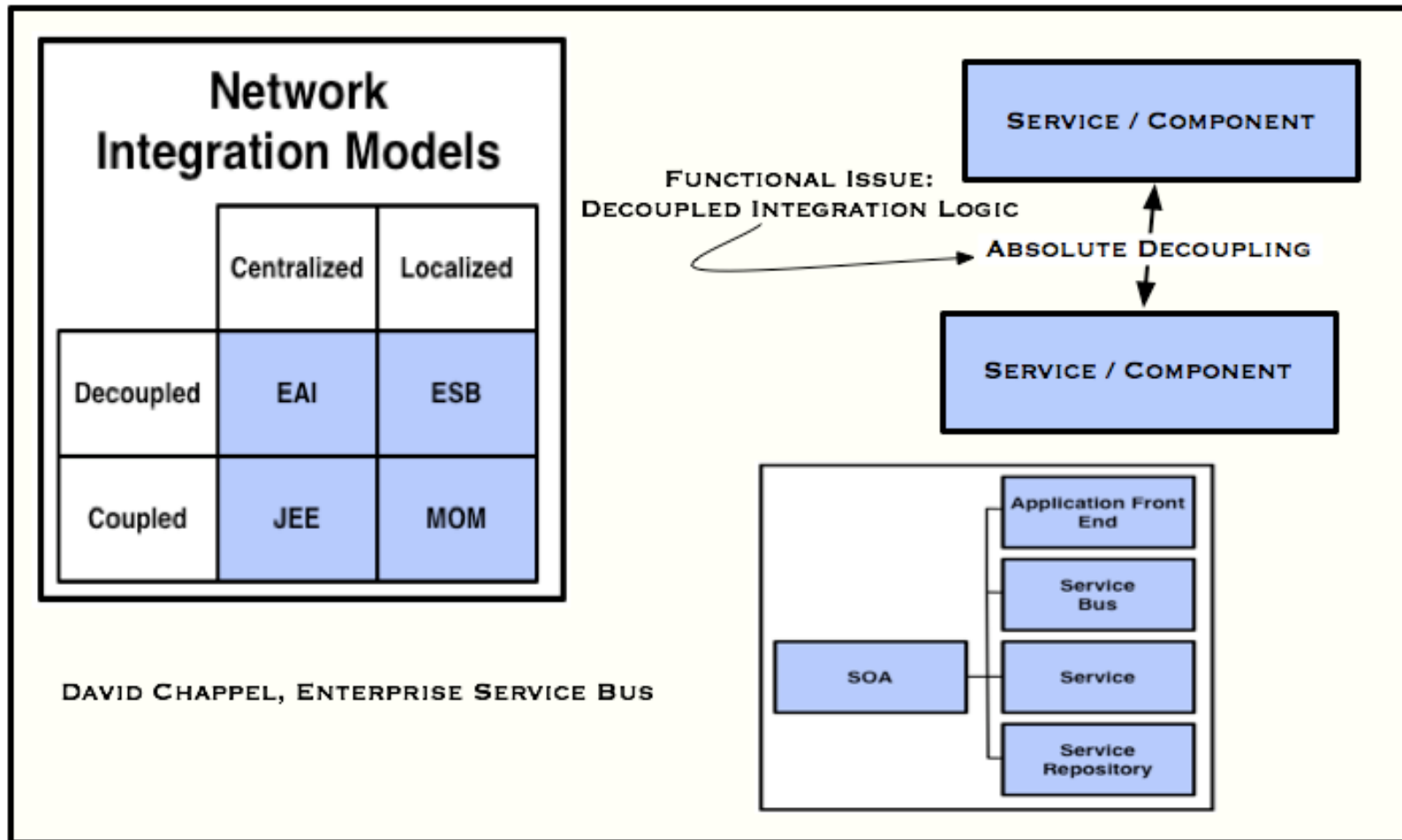
Topic: The use of mobile object technology in a net-centric system



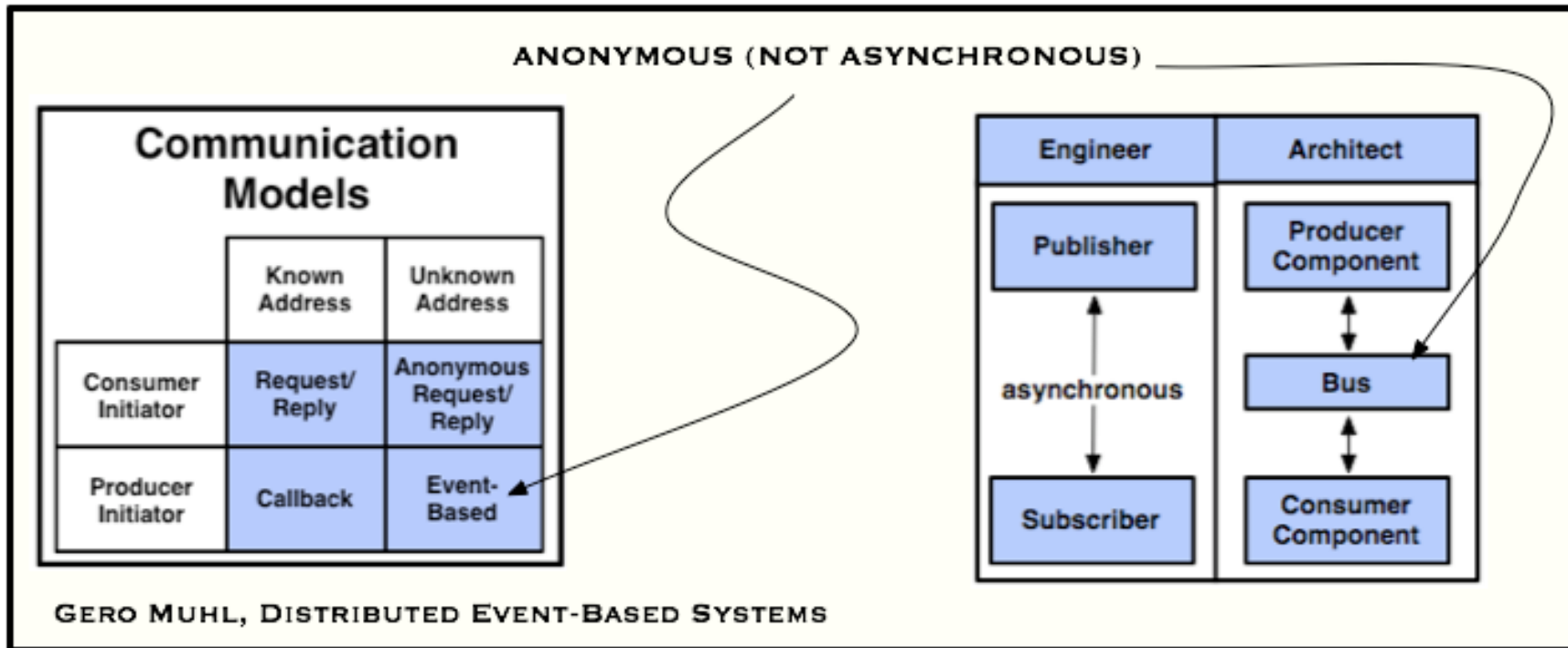
SOA fundamentals



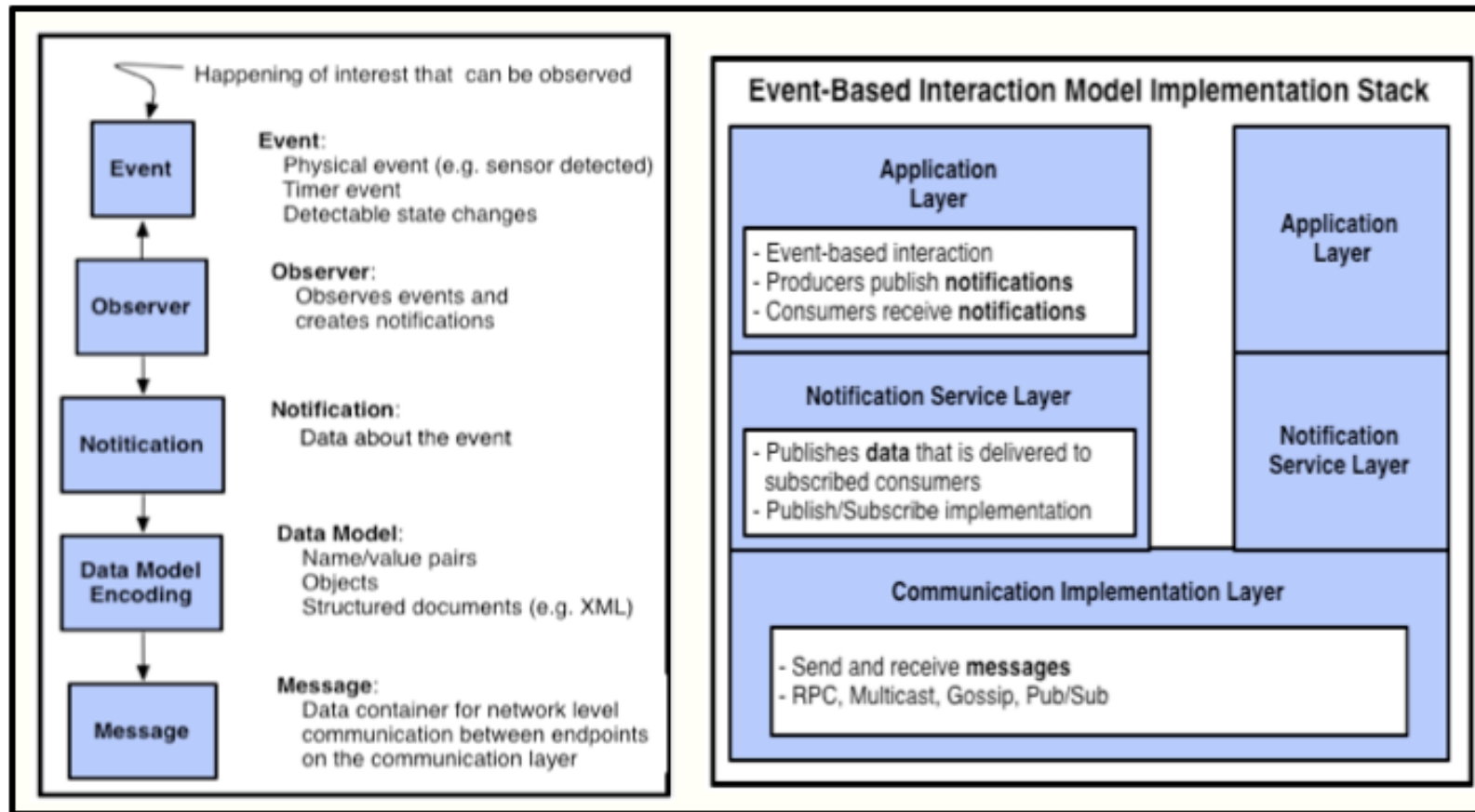
Network integration models



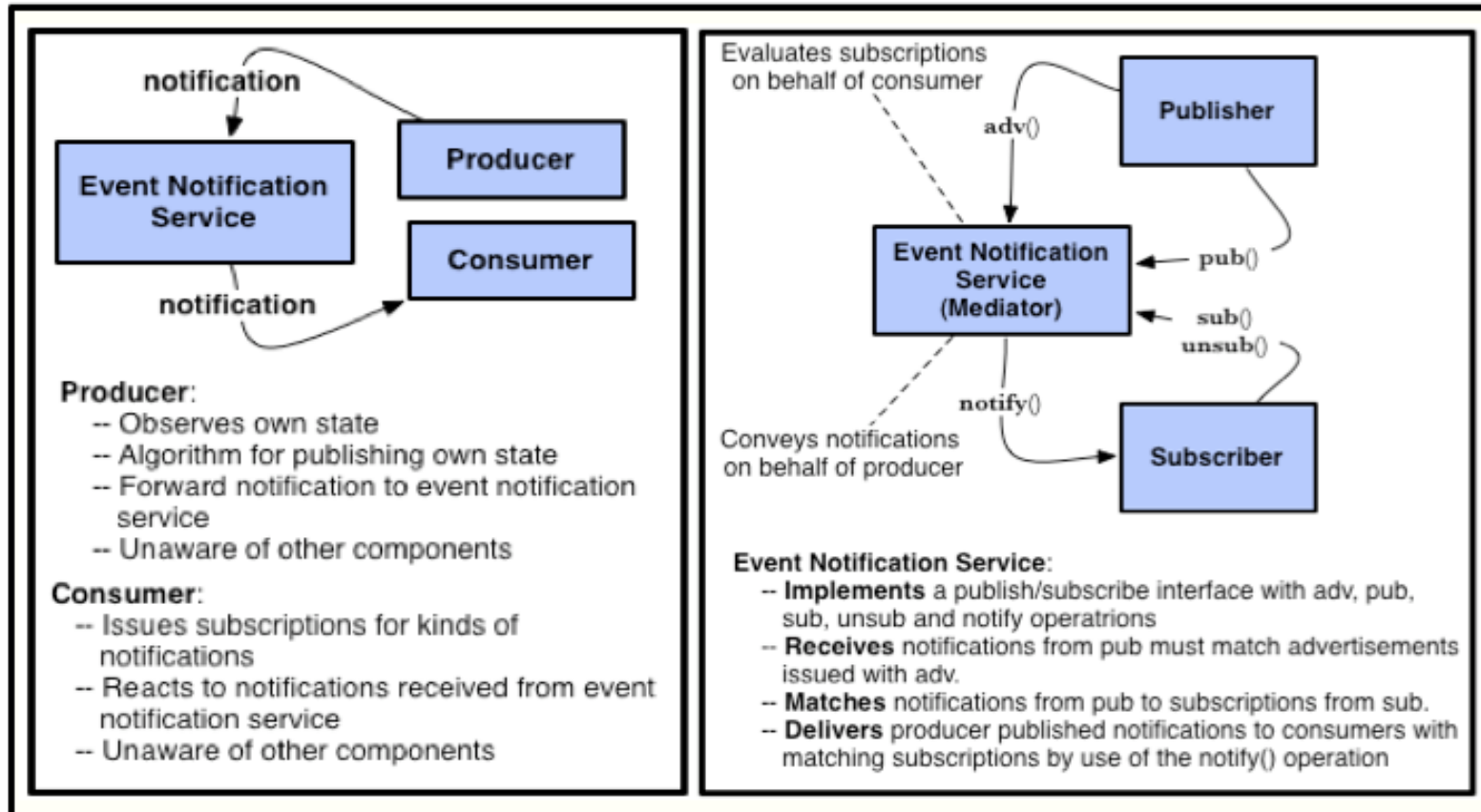
Communication models



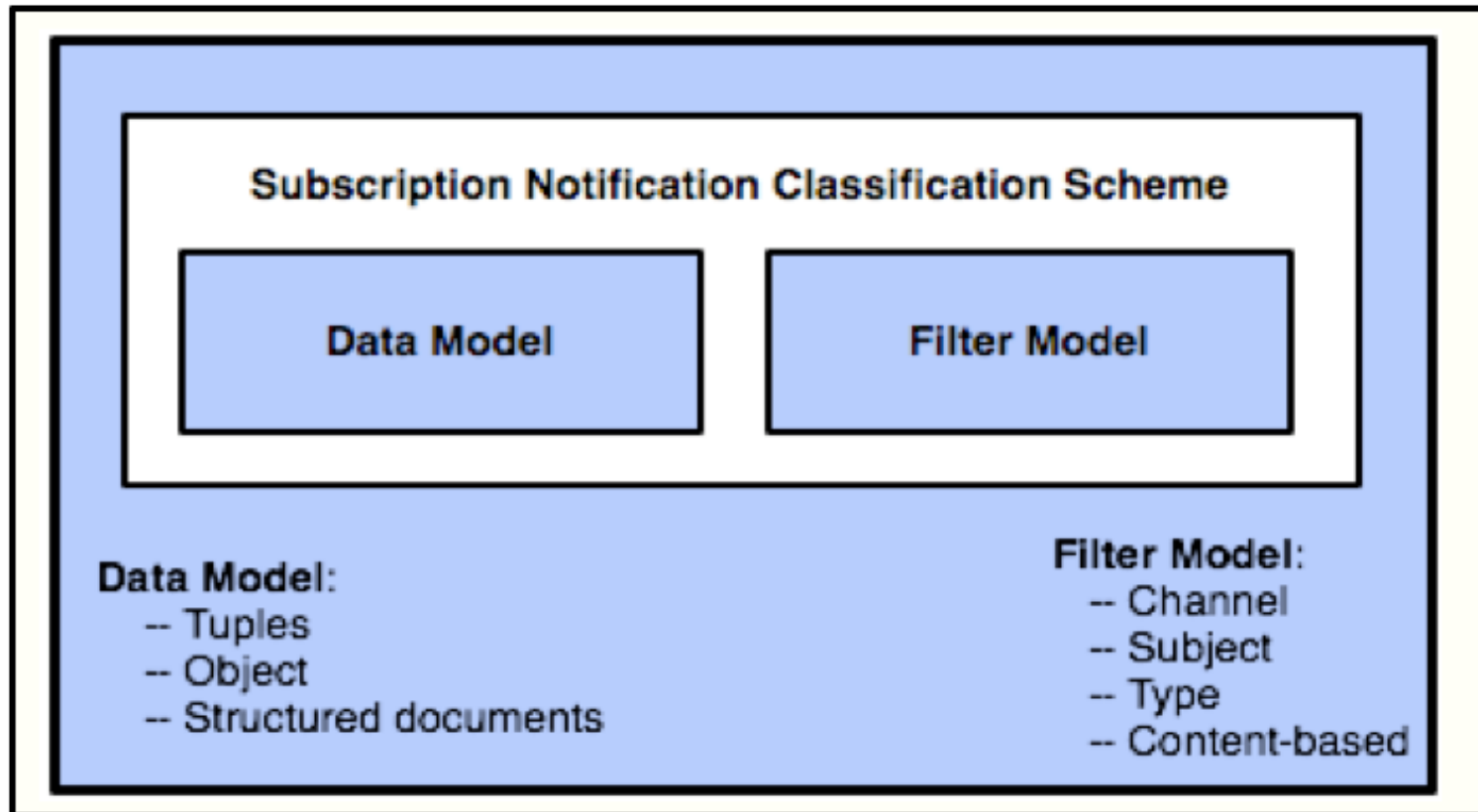
Event-based notifications



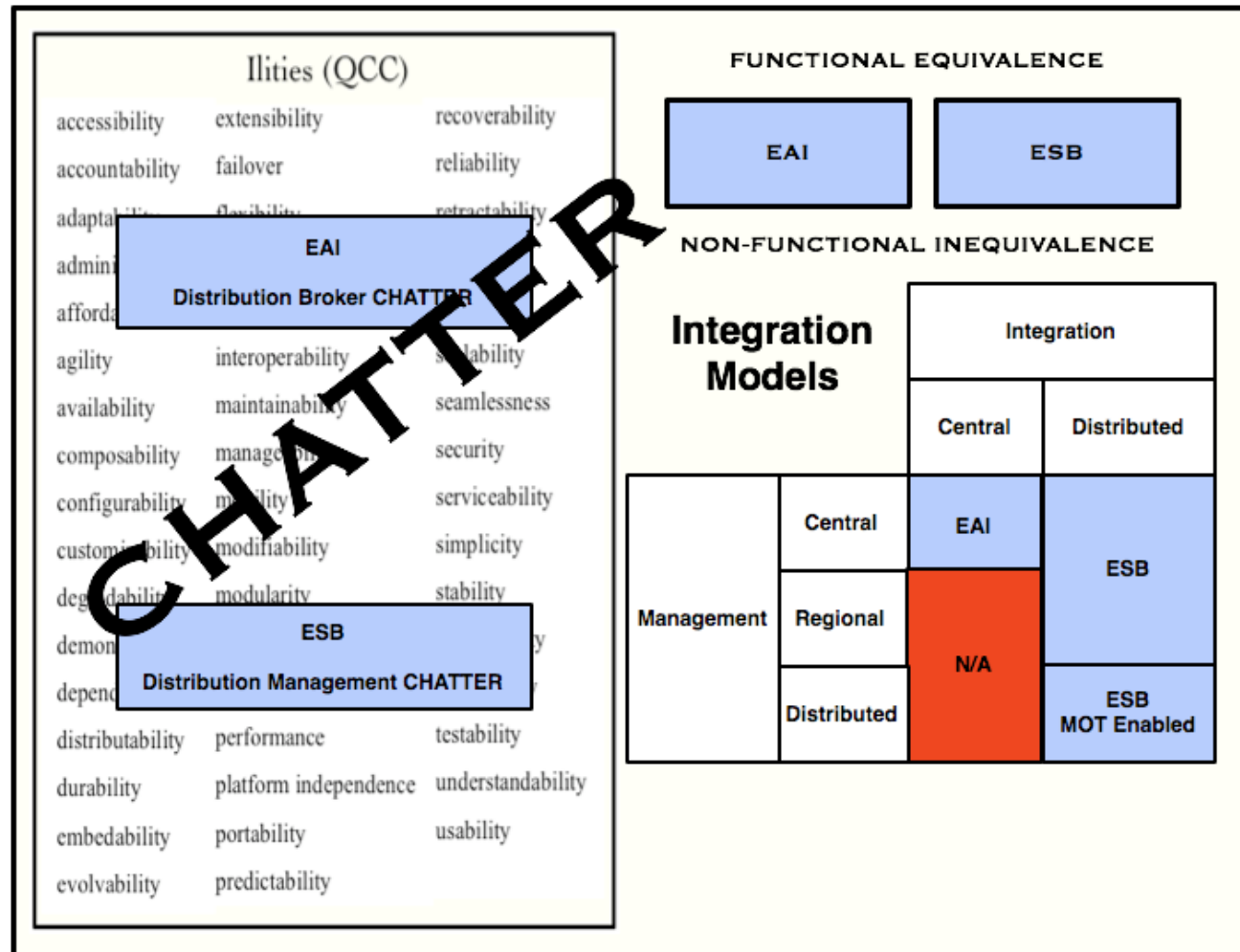
Event notification service



Subscription classifications



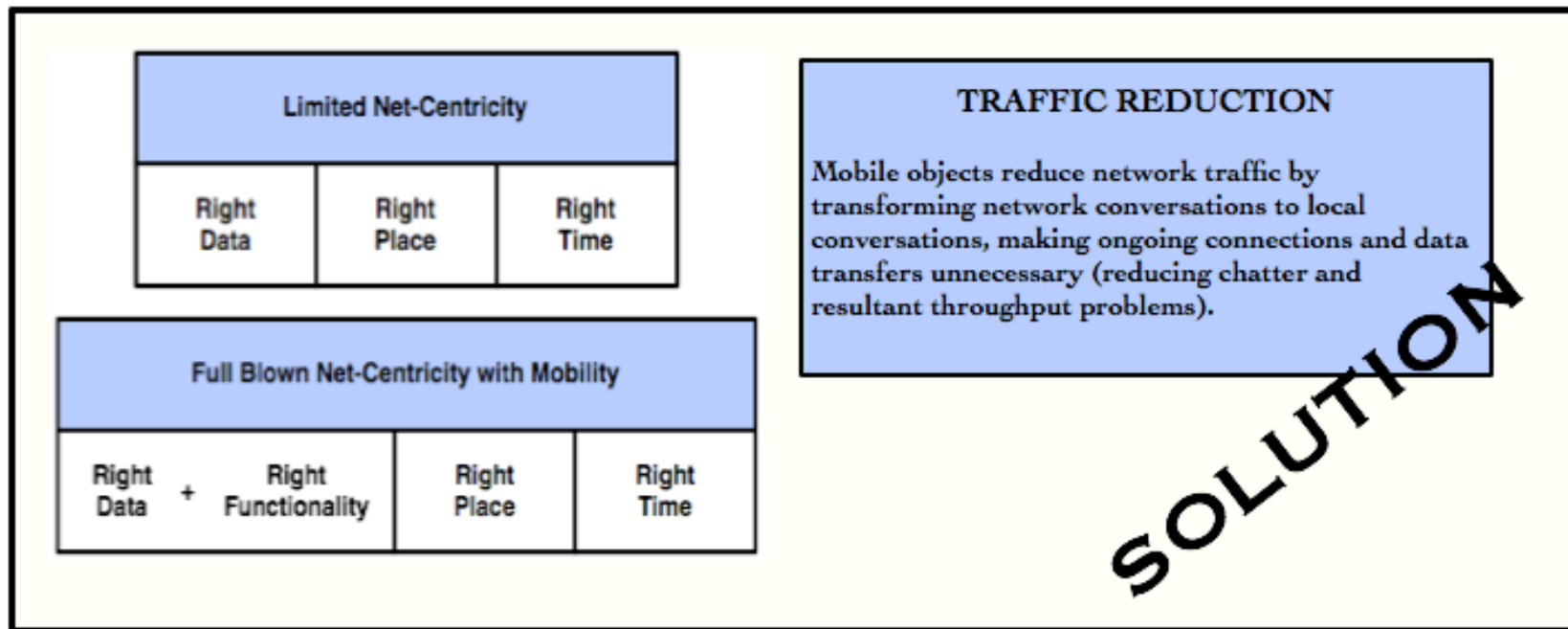
ESB chatter



Mobile object technology advantages

<p>Traffic Reduction</p> <p>Mobile objects reduce network traffic by transforming network conversations to local conversations, making ongoing connections and data transfers unnecessary (reducing chatter and resultant throughput problems).</p>	<p>Autonomous Decisions</p> <p>Mobile objects can sense their execution environment and react autonomously to changes. Multiple mobile objects, for example, can configure themselves in a network to optimally provide a solution.</p>	<p>Mediation and Collaboration</p> <p>Mobile objects can use a secure, trusted third party host for collaborative purposes, if they do not trust one another.</p>
<p>Throughput Reduction</p> <p>Mobile objects reduce network load by sending processing to where the data is rather than sending the data to where the processing is.</p>	<p>Seamless Integration</p> <p>Mobile objects provide for seamless system integration, since mobile objects are computer and transport independent (they live within the mobile object runtime environment).</p>	<p>Workflow Independence</p> <p>Mobile objects as workflow items can embody the information and behavior they need to move through an organization independent of any particular application</p>
<p>Protocol Translation</p> <p>Mobile objects can provide evolving adapters that handle legacy code protocol evolution without disturbing the network system.</p>	<p>Robust and Fault-Tolerant Builds</p> <p>Mobile objects make it easier to build robust and fault-tolerant due to their ability to react dynamically to unfavorable situations.</p>	<p>Leave the Nest</p> <p>Mobile objects can do their remote work without a need to keep contact with their original (creation) home.</p>
<p>Fragility Survival</p> <p>Mobile objects thrive on fragile networks. Mobile objects operate asynchronously and autonomously. Lost of connections does not affect mobile object code.</p>	<p>eCommerce</p> <p>Mobile objects provide real-time access to remote resources and, so, are well suited to eCommerce.</p>	<p>Next Generation</p> <p>Mobile objects can exceed the lifespan of their creator processes.</p>

Reduce chatter



Why?

WHERE ARE MOBILE OBJECTS THEN?

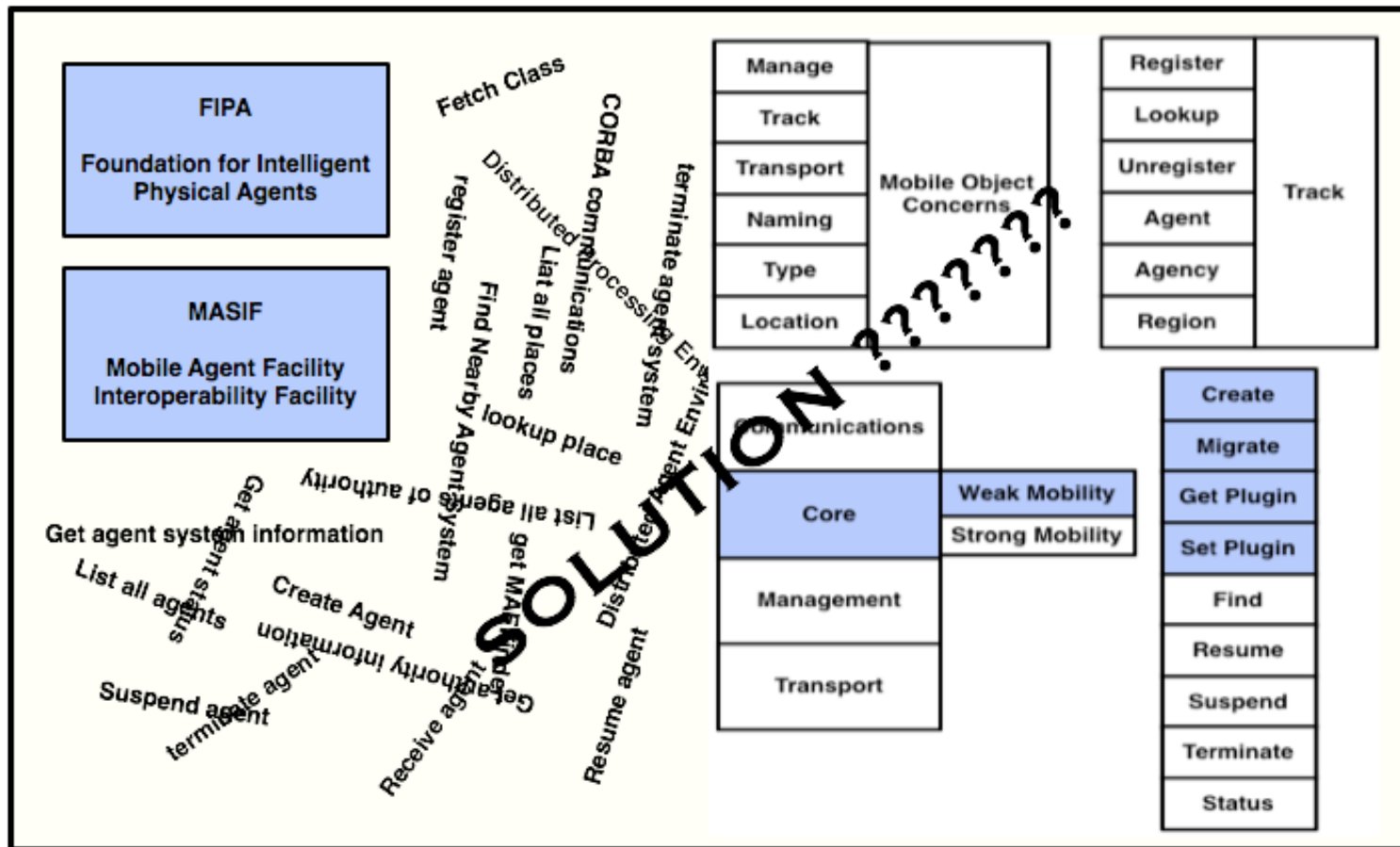
**WHY ARE MOBILE OBJECTS NOT
UBIQUITOUS RATHER THAN MORIBUND?**

Complex standards

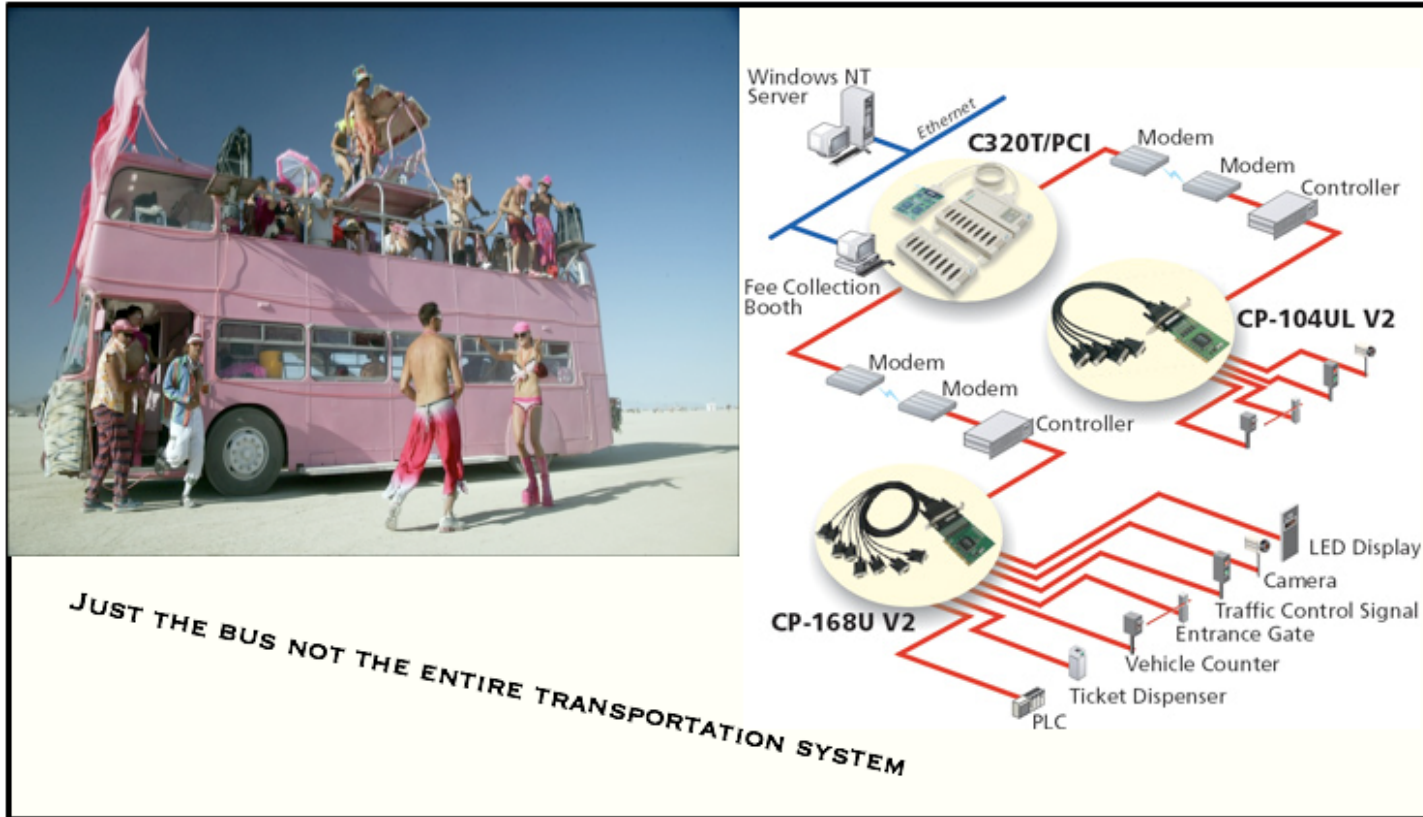
THEY ARE STUCK IN COMPLEXITY PRISON



FIPA and MASIF



KISS



Example KISS standard

Mobile Object Platform Core Standard Interfaces

```
public interface MobileObject extends MobilePlugin, Runnable
{
    void init ( Object [] parameters ) ;
} ////8-)

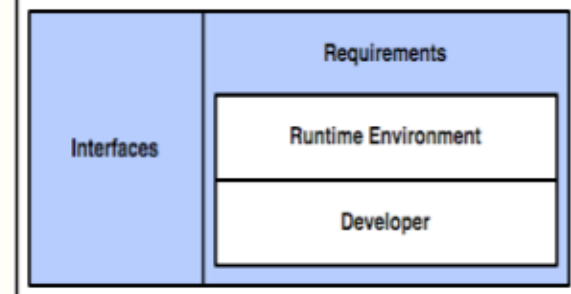
public interface MobilePlugin extends Serializable
{
    void setMobilePlatform ( MobilePlatform mobilePlatform ) ;
} ////8-)

public interface MobilePlatform extends Serializable
{
    void create ( String mobileObjectClass, Object [] parameters, File file ) ;
    void migrate ( String location, MobileObject mobile ) ;
    MobilePlugin getMobilePlugin ( String name ) ;
    void setMobilePlugin ( String name, MobilePlugin mobilePlugin ) ;
} ////8-)
```

JUST THE BUS NOT THE ENTIRE TRANSPORTATION SYSTEM

CREATE
INITIALIZE
MIGRATE
PLUGIN TO APPLICATION
RUN

Core Mobile Object Standards



Micro (modular) standards

