



Trenton Computer Festival

Credit Suisse IT Research & Development Innovations

33rd Annual Trenton Computer Festival - Professional Conference

Service Oriented Fabric A Data Center Clean Slate

Date: April, 26th, 2008

Produced by: "Head" Bubba

IT Research & Development

Credit Suisse

contact: Head.Bubba@credit-suisse.com

Head.Bubba@ieee.org

The materials may not be used or relied upon in any way.

Data Center Clean Slate → Research Seminar

Who Is Presenting – Credit Suisse IT R&D

- As one of the world's leading banks, Credit Suisse provides its clients with investment banking, private banking and asset management services worldwide. Credit Suisse offers advisory services, comprehensive solutions and innovative products to companies, institutional clients and high-net-worth private clients globally, as well as retail clients in Switzerland. Credit Suisse is active in over 50 countries and employs approximately 45,000 people. Credit Suisse's parent company, Credit Suisse Group, is a leading global financial services company headquartered in Zurich. Credit Suisse Group's registered shares (CSGN) are listed in Switzerland and, in the form of American Depositary Shares (CS), in New York. Further information about Credit Suisse can be found at www.credit-suisse.com.

■ IT Research and Development

- We look/develop/partner on the creation of new technology to solve our future business needs
- We have a lab in which we test new innovative technology with real applications

■ Who AM I? I am not Elvis, and please don't ask me to sing

The "Head" Bubba has 21 years experience in IT. I have been at Credit Suisse for the past 10 years, and have been a member of IT Research and Development since its inception where I am currently heading up Systems & Network Fabric Research; before joining R&D, I was a member of the Investment Banking CTO office working on high availability solutions along with operating systems enhancements. I started out as a developer for Equity trading floor applications. Prior to Credit Suisse, H.B. was an independent consultant designing systems using object oriented design methodologies. The "Head" Bubba has six Jack Daniel's Single Barrel Whiskey Bottles, and gives 90% of the bottles away to individuals who create innovative new technology. Visit www.HeadBubba.com for more info on H.B.

Why We Agree With Stanford's Clean Slate Program Goals



Service Oriented Fabric

Future Trend – Clean Slate

- **Program Goals:** We believe that the current Internet has significant deficiencies that need to be solved before it can become a unified global communication infrastructure. Further, we believe the Internet's shortcomings will not be resolved by the conventional incremental and 'backward-compatible' style of academic and industrial networking research. The proposed program will focus on unconventional, bold, and long-term research that tries to break the network's ossification. To this end, the research program can be characterized by two research questions: "With what we know today, if we were to start again with a clean slate, how would we design a global communications infrastructure?", and "How should the Internet look in 15 years?" We will measure our success in the long-term: We intend to look back in 15 years time and see significant impact from our program.
- **We believe researchers and technology firms need to look outside the box, and stop concerning themselves with breaking a standard**
 - **Distributive technology like Linux and Infiniband & RDMA have proved there are people who will adopt the new technology as well as demonstrating innovation will win the end over traditional solution – *it just takes too long right now***
 - **What we see 95% of the time is the capitalization on the short comings on the current network in the data center, but never offer real improvements**

Service Oriented Fabric

Prologue

- **This presentation takes into account the notion of the program goals of the Stanford Clean Slate program, and applies them to a data center to discuss a possible new solution to solve data center issues we face today & tomorrow. Pre-conceived notions of how things are done today are dismissed, and starts with a clean slate. During some of the original presentation people began to apply today's technology to a potential future solution that is a concept, and did not fully take into consideration a clean slate.**
- **Every business sector has different needs, and within a data center each line of business can have different needs**
 - **One can not simply take the model of any business sector and apply it to another; for example, an airport or tracking of packages. In terms of market data to handle miss-ordering, firms have two redundant feeds coming in which doubles the requirements; this creates programs more complex as they have to attempt to fill gaps across two feeds to see if something should be rejected or not rejected - if they can find a missed message in the second feed. *Tolerance of margin of error in processing is different for each line of business not just a sector.***
 - **For example, when I state can't have, I list miss-ordering. To latency sensitive trading applications miss-ordering = lost message(s).**



The Problem

Service Oriented Fabric

The Problem –One Size Does Not Fit All

- **Whenever something new is introduced there is the age old fight against what is new**
- **Resistance to change is high, and it is like a fight to the death to introduce something new even if it is mandated as the strategic direction**
 - **This effects Research projects as something new won't be discovered as it won't be allowed to be looked into**
 - **Enterprise business customers suffer as they don't get the latest and greatest**
- **Then The Worst Of The All, Just Because Only One Entity Innovates Something That Adds Benefits Gets Dismissed Solely Because The In-House System Doesn't Do It And "The Followers" Say It Is Not Needed**
 - **In 1999, UNIX Technology that improved performance & high availability that did sub-second failover were dismissed by most in the industry due to incumbent footprint, and unfortunately the technology took 2-10 minutes and needed application re-start.**

Service Oriented Fabric – A New Concept To Consider

Service Oriented Fabric

Future Trend – Untraditional Network Research

- **The Need For Something New in Networking Technology**
 - The network was previously incorrectly thought only as plumbing, but its removal from application needs has failed; with ever increasing volume of messages/sec and/or data sent to be processed, the network must evolve into something new...
 - By creating a *Service Oriented Fabric (SOF)*, the network becomes a platform resource with an increased amount of functionality, and oriented around application requirements that can be virtualized by allocating dedicated bandwidth (highway lanes or rails)
 - Collapses monolithic systems into modular distributed systems
 - Reduce operational complexity by orders of magnitude
 - ↳ Reduce “JITTER” (not just a OS thing, the network does have it, some people will not consider and/or admit network “Jitter” exists and/or is an issue
 - ↳ System, OS, Network, Storage etc... all becomes objects in a fabric that can be virtualized

Service Oriented Fabric

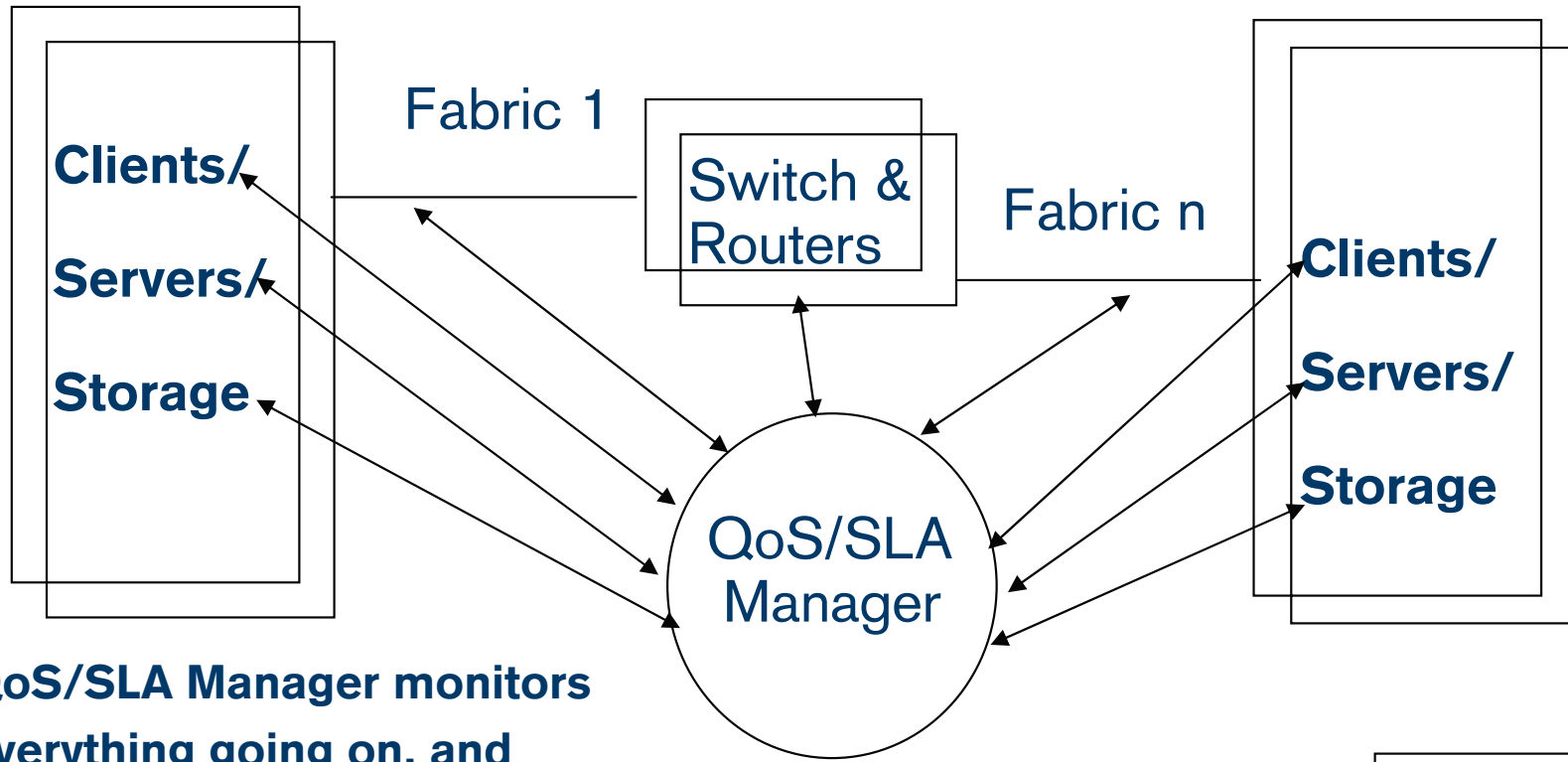
Convergence = Consolidation + Virtualization

- **All traffic on one Fabric: LAN + FC/StAN + IPC/SAN =>**
 - **Virtually, but safely, separated traffic:**
 - Trading
 - Storage (FCoE and RDMA / iWARP)
 - Customer login (secure access to internal network)
 - Development / R&D
 - Management
 - **Quality of Service (QoS) Provisions**
 - No Loss (Storage and IPC)
 - Delay guarantees for trading traffic (and customers)
 - Bandwidth guarantees for storage
 - **Congestion Management (CM) and Load Balancing**
 - under application control (API)
 - No “PAUSE” sent without consideration of applications Service Level Agreement (SLA)
 - **Deadlock Management (DM)**
 - DLKs are more catastrophic than congestion => must have solution
 - **Destination (DST)-based per-flow RX rate calculation (throughput accounting)**

Management Of The Service Oriented Fabric

Service Oriented Fabric

What Is It? – Virtualize Everything!



- **QoS/SLA Manager monitors everything going on, and makes real time adjustments including live migration of an application or family of applications**

- **Everything is virtualized!**
- **Dynamic/Changeable Dedicated Bandwidth makes it happen !**

Real Time Data Center Power Cooling Monitoring and Adjustment System

Service Oriented Fabric

Business knows the service needed, not the systems

■ Service Oriented Computing (SOC)

- Requires an integrated approach to simplify entire service stack (application services and infrastructure) with its focus on rapid service and capacity delivery to the business
- Application services conform to *Service Oriented Architecture (SOA)*, with platform and language independent framework providing a basis for enterprise deployment, service management and service implementation
- Infrastructure layer is ‘virtualized’ - *Service Oriented Infrastructure (SOI)* that can be flexibly provisioned based on service level objectives
 - Everything is virtualized! -> *including*
 - ↳ Servers / Operating Systems
 - ↳ Storage
 - ↳ Network became virtualized to become a *Service Oriented Fabric (SOF)* with dedicated bandwidth

Service Oriented Fabric

Application Migration Considerations

■ Stateless Application Vs Statefull Application

- A Stateless application has no “state” and can be stopped and re-started without impact on any server
- A Statefull Application has data, and needs that data to continue to process

■ Moving applications in a data center need to consider these two types of applications

- *An application may be dependant on one or more applications*
 - ↳ *If one needs to be moved, to continue to meet the SLAs you may have to move all the applications in what we will call fabric “cluster family”*
- *You can not depend on server resources alone to determine where to run and where not to run – PLUS virtualization complicates matters*

Service Oriented Fabric

Management Of The Fabric

■ Quality of Service (QoS) / Service Level Agreement (SLA) Manager

- Application performance requirements monitored in real time → ex:

- ↳ CPU utilization

- ↳ Memory utilization

- ↳ Service Oriented Fabric Latency & Performance (*remember network evolved to this!*)

- Combining a Logic Machine Architecture (LMA) with an Interactive Theorem Prover (ITP) opens the door for real time proofs

- determine ahead of time if an object in an application is approaching its overload

- ↳ Its not about a server or application: Interaction with Real Time Data Center Monitoring system is essential to determine where to move an application in the data center should the application coming close to running out of a resource be moved should other applications be migrated to avoid moving a family of applications and what part of the data center has the power/cooling requirements

- Now you have the justification to create a Service Oriented Fabric, so now let's take a deeper dive...

Service Oriented Fabric

QoS/SLA Manager

■ Example of possible generated logic from a sample proof:

- If (!\$SLA (CPU_UTILIZATION) or !\$SLA(MEMORY) or !\$SLA(MEMORY_BANDWIDTH) or !\$SLA(FABRIC_IN_LATENCY) or !\$SLA(FABRIC_OUT_LATENCY) or !\$SLA(STORAGE_IN_BANDWIDTH) or !\$SLA(STORAGE)_OUT_BANDWIDTH) or !\$SLA(SYSTEM_PERFORMANCE) Then \$DISCOVER_FABRIC(LIVE_DATA_MIGRATE)

↳ SLA is a function that checks the resource id passed to it

↳ Another example may just be: If (!SLA(TRANSACTIONS_SEC) Then FABRIC_BANDWIDTH_REALLOCATION(REDUCE_BANDWIDTH(1, 2.0) && INCREASE_BANDWITH(2, 5.5))

↳ Discover Fabric is a function that finds a location for the application to run or if it's a family of applications where the all the applications should run as we don't want to break SLAs of latency of one application to another; it interfaces with the Data Center Monitoring and basically multiple proofs are run to discover the best set-up in the data center and if any power/cooling changes need to happen the Data Center Monitor will ensure that is adjusted (ex: where the move is happening may require an increase in power/cooling, and we may be able to save money by educing power/cooling after the live migration of the application or application family)

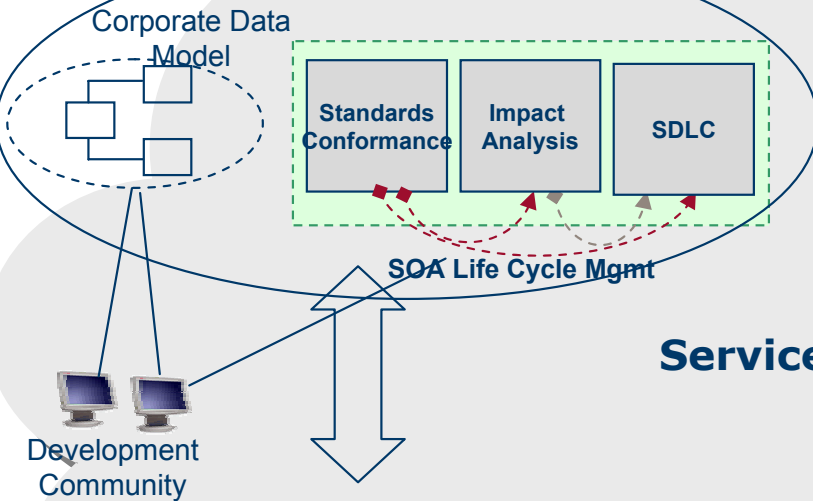
- *We can discuss more types of proofs and functions in Q/A if you want to! And even how you could take source code for Prolog++ to create a LMA/ITP application to create a QoS/SLA Manager*

Service Oriented Fabric

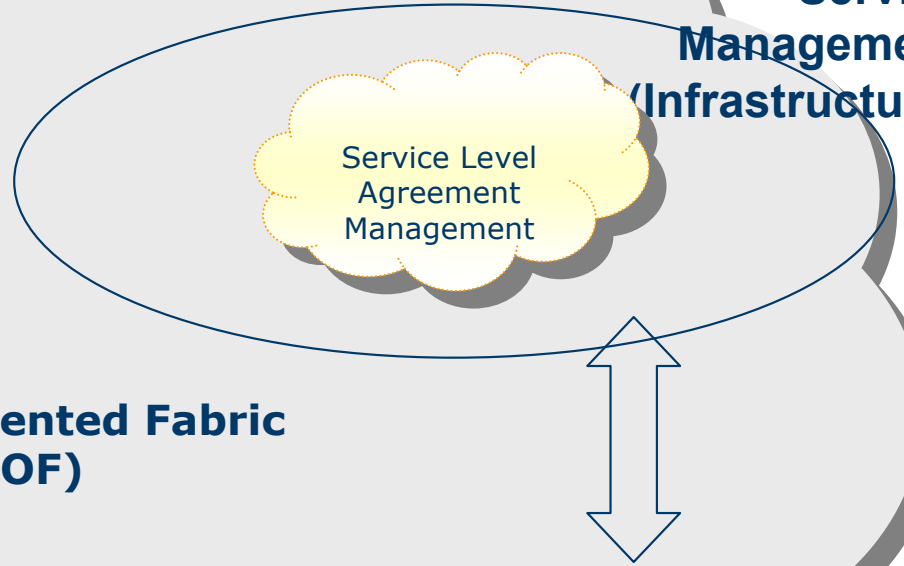
Everything Based On Open Standards

The Elements of Re-Invention

SOF Governance

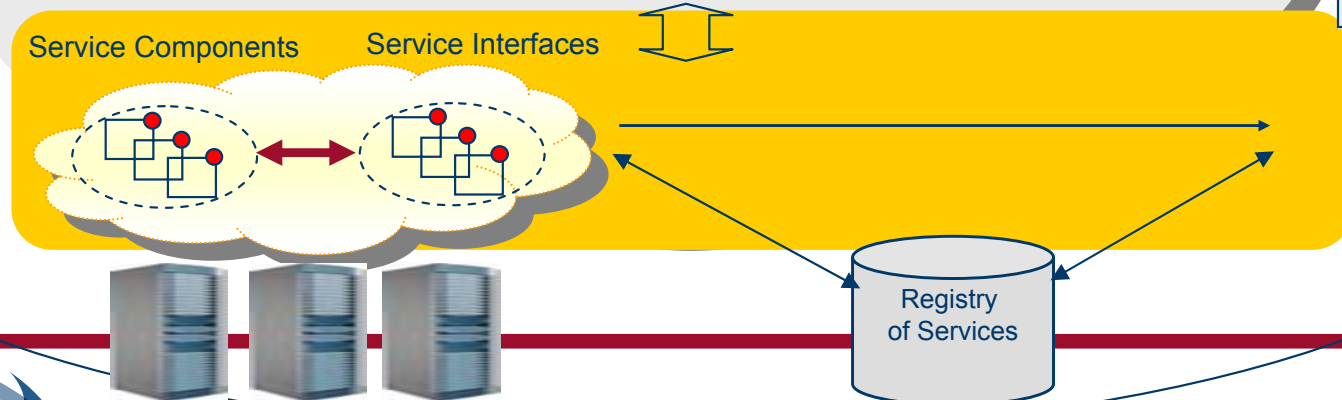


Service Management (Infrastructure)



Service Oriented Fabric (SOF)

Application Services

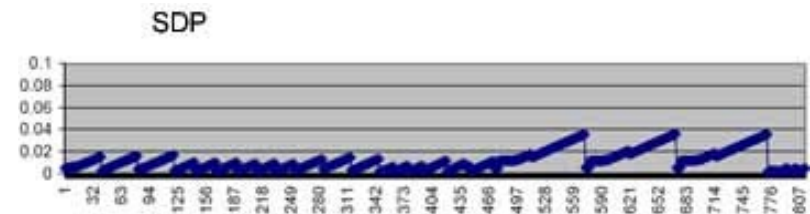
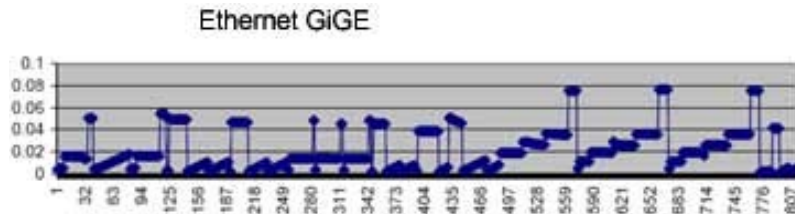


Service Oriented Fabric Requirements

Service Oriented Fabric

Requirement No Network Jitter

- *Requirement: Need consistent low latency predictable message delivery*
 - *Ex: Socket Direct Protocol (SDP) on Infiniband does not require special tuning*



- Need predictable message behaviour, low latency, high bandwidth

- Large-page support
 - ↳ Would help support many of the large data structures and buffers needed for supporting high-bandwidth networks
- RDMA – seen to have benefits
 - ↳ need stable interfaces, auxiliary interfaces (ex: pin memory that is subsequently unmapped), etc...
- TCP/IP - offloading has also seen to have benefits
 - ↳ *user space options emerging, so is it really an option? (with real time?)*

↳ See: <http://www.solarflare.com/technology/documents/EndoftheRoadforTCPOffload.pdf>

- Scalability of protocol processing over multi-core architectures
- Examples of some initial testing (not finished, - this is just one set of tests at a point in time)

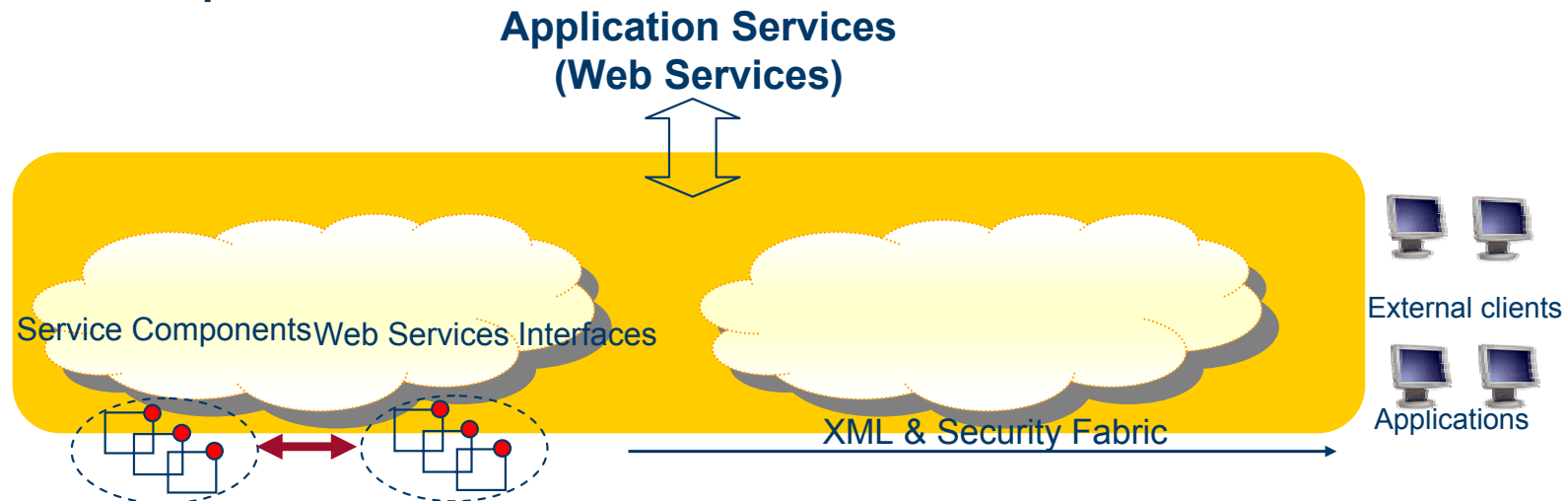
NIC Benchmark using NetPipe ping-pong mode:

| | NIC1 | NIC2 | NIC3 | NIC4 | NIC5 | NIC6 | iWarp1 | iWarp2 |
|----------------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| 64-byte NetPipe Latency | 15us | 24us | 28us | 100us | 30us | 32us | 8us | 7us |
| Highest NetPipe throughput | 5.3Gbps | 4.7Gbps | 4.4Gbps | 1.4Gbps | 5.4Gbps | 3.4Gbps | 8.7Gbps | 8.9Gbps |

Service Oriented Fabric

Enterprise XML and Security Fabric

- “XML Fabric”
- SOA governance and WSM must be coordinated to address the issues of managing the “XML fabric” that compose the SOA framework
- XML will become so engrained within the communications fabric that it will deteriorate mandatory service levels! Create unachievable service levels!
- Must be treated as an evolving ecosystem, thereby imbedding acceleration seamlessly within the new network platform



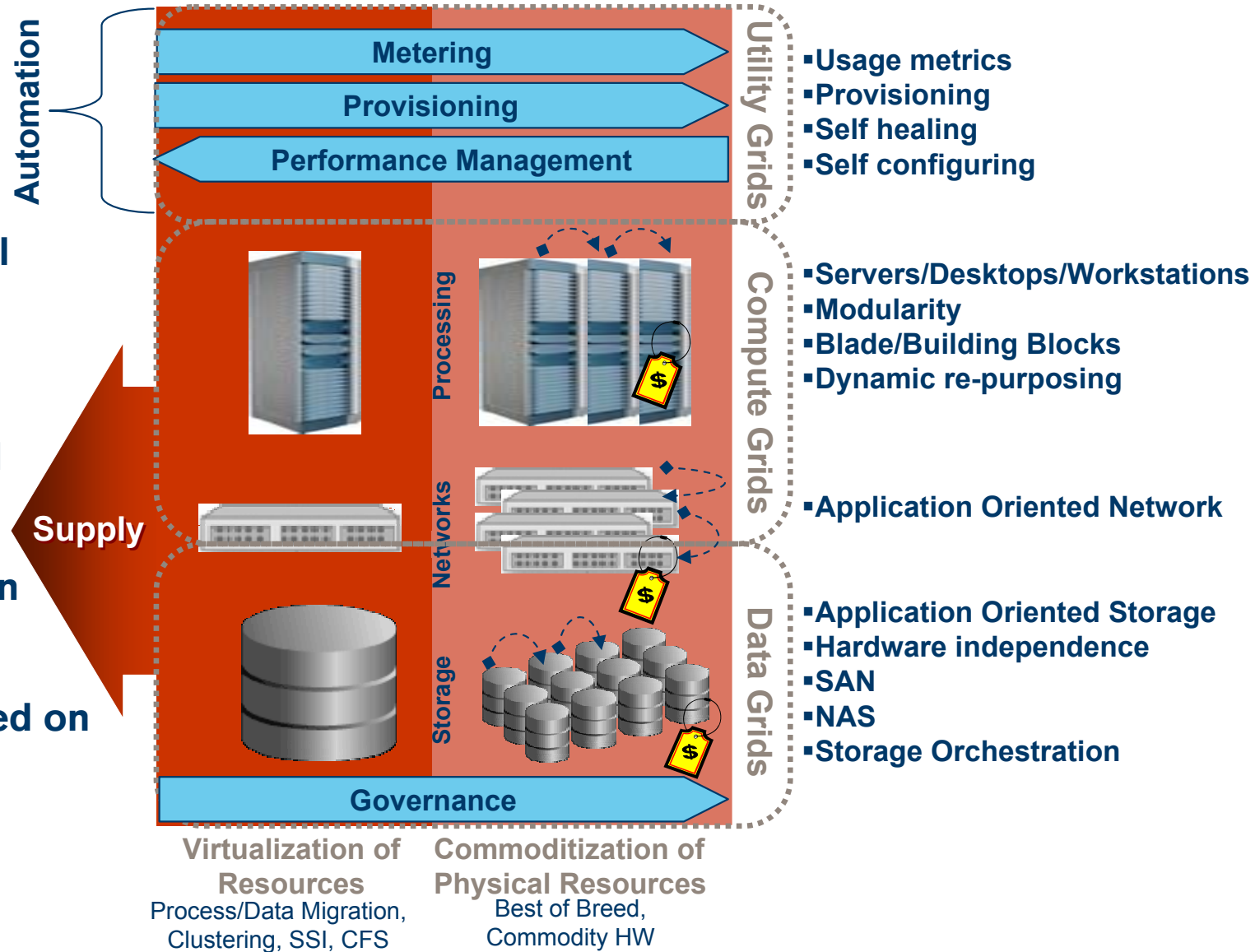
“Security Fabric”

- Security requirements must be treated holistically
 - Security management must be implemented and configured independently of application implementation
 - Security must be closely tied to contents of XML fabric
- Security must shift from the edge to embedded within the Service Oriented Fabric
 - To work like Human immune systems

Service Oriented Fabric

Infrastructure Ecosystem To Simplify IT

- Solves the operational complexity
- Drives up agility
- Solves network based computing
- Breaks the squeeze on innovation
- Delivers capacity based on demand



Enables business that differentiates on interoperability and/or performance

Service Oriented Fabric

Key Needs For Datacenter Applications

■ Must not have:

- Traffic mix using Priorities which create un-managed bandwidth which leads to congestion (different classes merged by switch or NIC)
 - Need to adhere to Service Level Agreements (SLAs)
- Link-level loss (CM is no match for a fast LL-FC)
- Mis-ordering
- Deadlocks & Congestion
- Congestion Control that can cause “stale” data by sending “PAUSE” ignoring SLAs
- SlowStart (ramp-up a la TCP)

■ Must Have:

- Standards based solution for above not haves - No propriety solution
- Dedicated *Highway Lanes* (or rails) for Bandwidth
- Clock Synchronization
- Measurement
 - Need to know what the latency and throughput are, and how they change over time
 - ↳ Each msg needs to have a timestamp on it as it passes through switch, router, servers & client
- Lossless – but not at the expense of possible “stale” data by SLA ignored via PAUSE

Service Oriented Fabric

Alternative Exemplary DC-ICTN Exists

- Infiniband can & meets many requirements
 - Lossless LL-FC
 - 15 Virtual Lanes
 - 0-copy, low-latency possible
 - Native RDMA based on VIA (Queue Pairs)
 - 10 / 30 Gbps at lowest cost/port...
 - DDR doubles this!
 - Claims Congestion management was an afterthought in an addendum are not true
 - Applications will be able to control Bandwidth Allocation Of Virtual Lanes
 - API and XML to control Virtual Lanes by bandwidth to meet SLAs is being addressed for next generation
 - There will not be un-managed bandwidth as there is with Ethernet nor will there be a “PAUSE” to cause stale data

- Ethernet considering Bandwidth By Contract Period of Time for Video/Audio
 - This needs to be extended to un-limited period of time and not taken away if not used

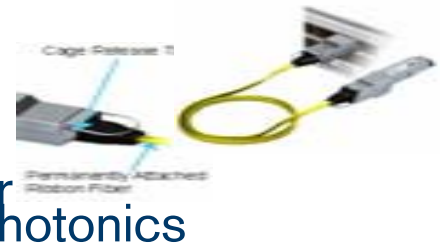
Service Oriented Fabric

Key Needs For Datacenter Applications

- **Why “PAUSE” can be bad if a SLA is ignored**
 - Consider a Market Data and Trading Floor Applications, with the ever increasing amount of messages/second latency is already becoming a major issue
 - The introduction of a fabric allowing a “PAUSE” to be sent will cause market data to become stale without the application knowing it
 - ↳ Wrong trading decisions will happen
 - ↳ Money will be lost
 - ↳ This will affect your 401k and retirement funds – do you want to lose money because some one took a short cut and didn’t consider “stale” data
 - There are some that want to take the easy way out and send a “PAUSE” without putting in Management into the fabric to enable SLAs to be used to determine if a “PAUSE” is allowed for an application – this why I got invited to present at 802.1au
 - ↳ Note: Many got the message when I presented in July at 802.1au, but some still don’t care
 - Now, you know why you need managed bandwidth with allocated bandwidth for applications to use. Now let’s consider a Clean Slate Approach for the Data Center of the future as well as sending data from one application to one or more applications

Building Blocks & Virtualization Of The Entire Data Center – Sample Use Of A Service Oriented Fabric

Example Of A Building Block Reference Architecture

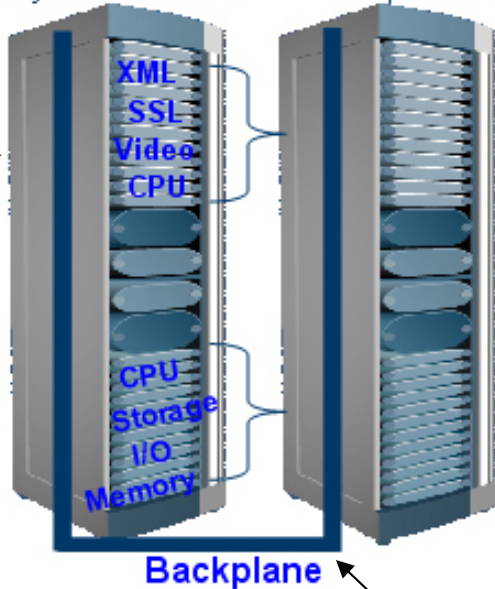


Data Center Photonics

Infiniband over Photonics (IBoP) and Ethernet over Photonics (EoP)
-or- new Photonics protocol

Aggregate "Appliance-Like" Building Blocks

Policy Based Server/Desktop/Workstation



Optional Global Address Space For Memory In Each Server

Contiguous memory allocated at boot time
BUT Servers don't need to have memory

Memory Building Blocks with Optional Global Memory System

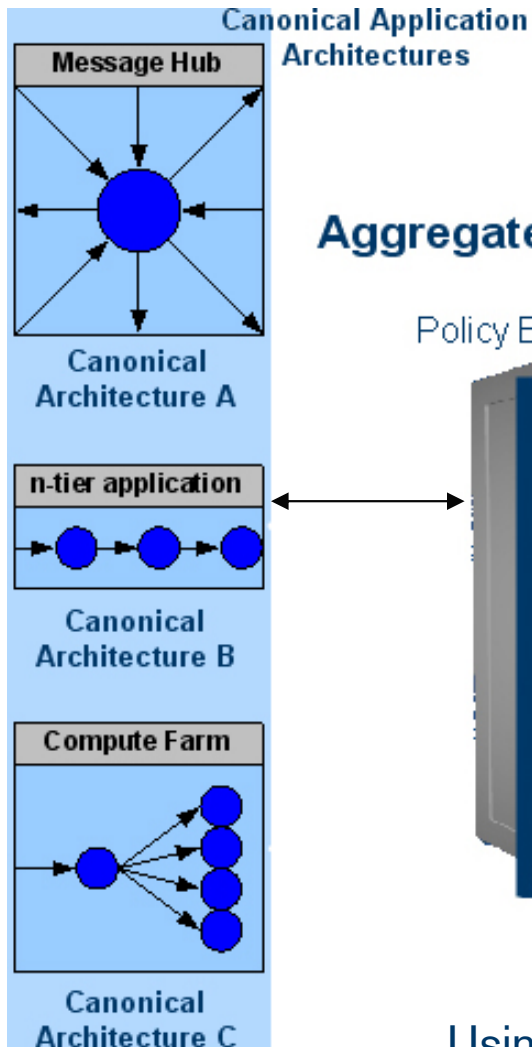
When used blades can maybe 1 TB of Memory

CPU Building Blocks Storage Building Blocks



Memory Channel like API consideration for OFA by Head Bubba
RDMA / iWARP -or- new
Silicon Photonics technology platform verbs
-or- new verbs

Photonic Technology Backplane



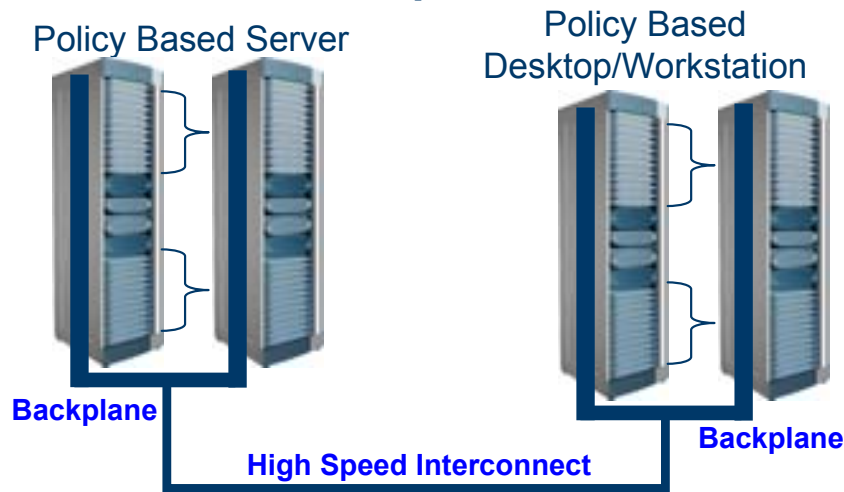
Each application classification is defined holistically by its needs, and built up from core building blocks

Using an Optical Interconnect we create the CORE building blocks of what we call a system today

Service Oriented Fabric

Building The Service Oriented Fabric

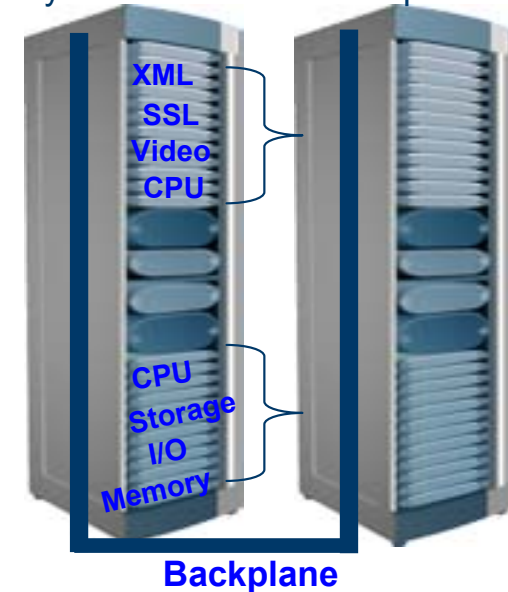
Building Blocks – Server/Desktop/Workstation



- Silo specific blades-desktop/workstation/servers
- Building blocks on CPU/I/O/network/storage
- Horizontal/vertical scaling
- Engineer away the complexity
- Solve power/space/heat
- Synchronizes application goals with platform service levels
- Network treated as a platform
- On-demand rip and replace

Aggregate “Appliance-Like” Building Blocks

Policy Based Server/Desktop/Workstation



- Embedded Fabric virtualization allows integrated desktop/workstation/servers
- Adding appliance based building blocks complete virtualization of compute fabric by providing specialized services on the same virtualized Service Oriented Fabric

Service Oriented Fabric

Lifecycle Management today

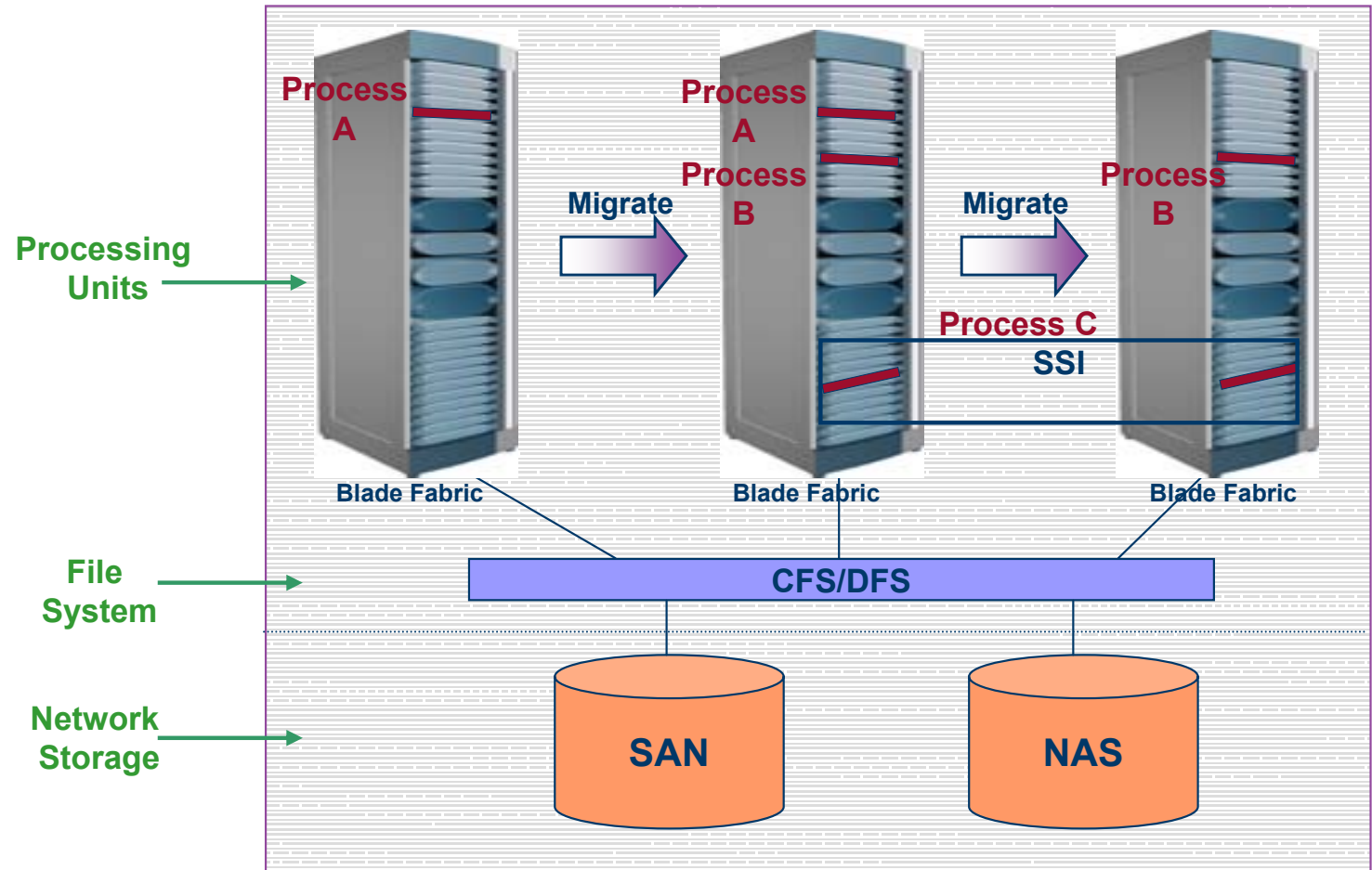
Accelerate ability to enable growth, lower cost & deploy faster

Service Oriented Fabric Lifecycle Management

Migration of Stateless/Statefull applications

- Process Migration
- Clustered File System
- DSM
- SSI
- VSAN
- High Speed Interconnect

Automated Provisioning/Orchestration

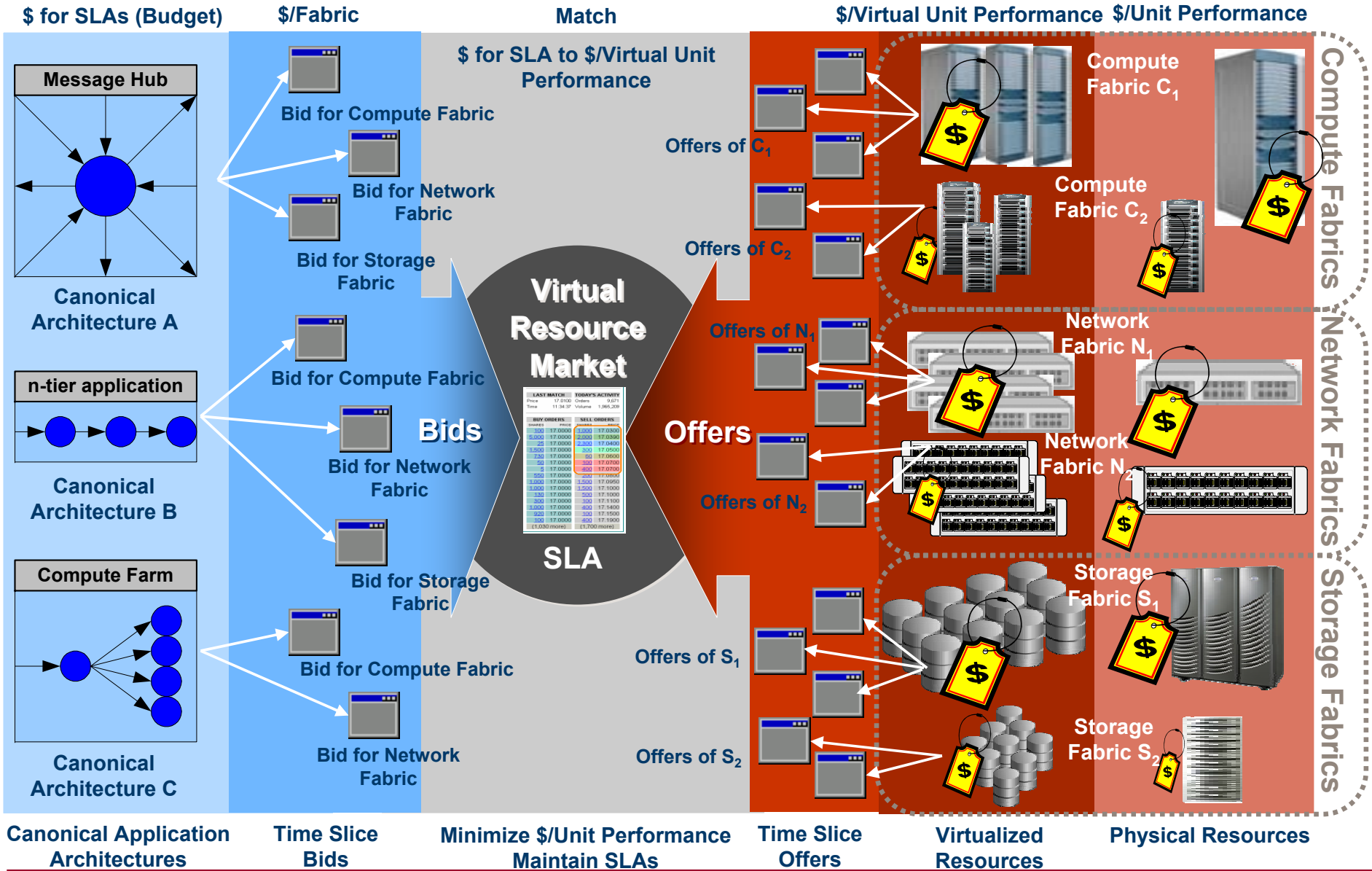


Fabric SMP

Virtual Machines will be the core container for migration and virtual SMP

Service Oriented Fabric

The Future: Lifecycle Management with A Virtual Resource Market



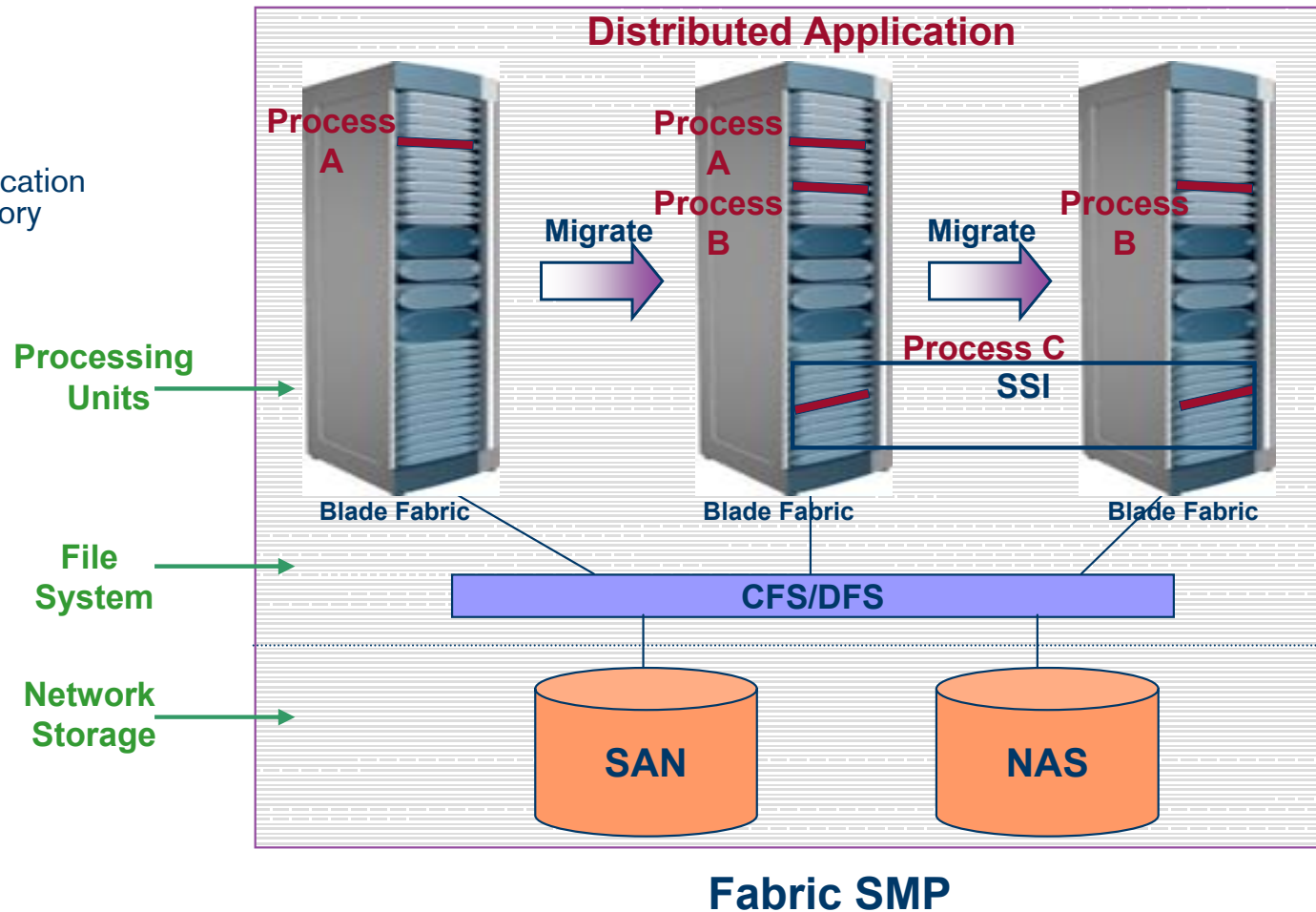
Service Oriented Fabric

The Future A Virtual Resource Market

Service Oriented Fabric enables creation of a Virtual Resource Market (VRM)

Core Building Blocks

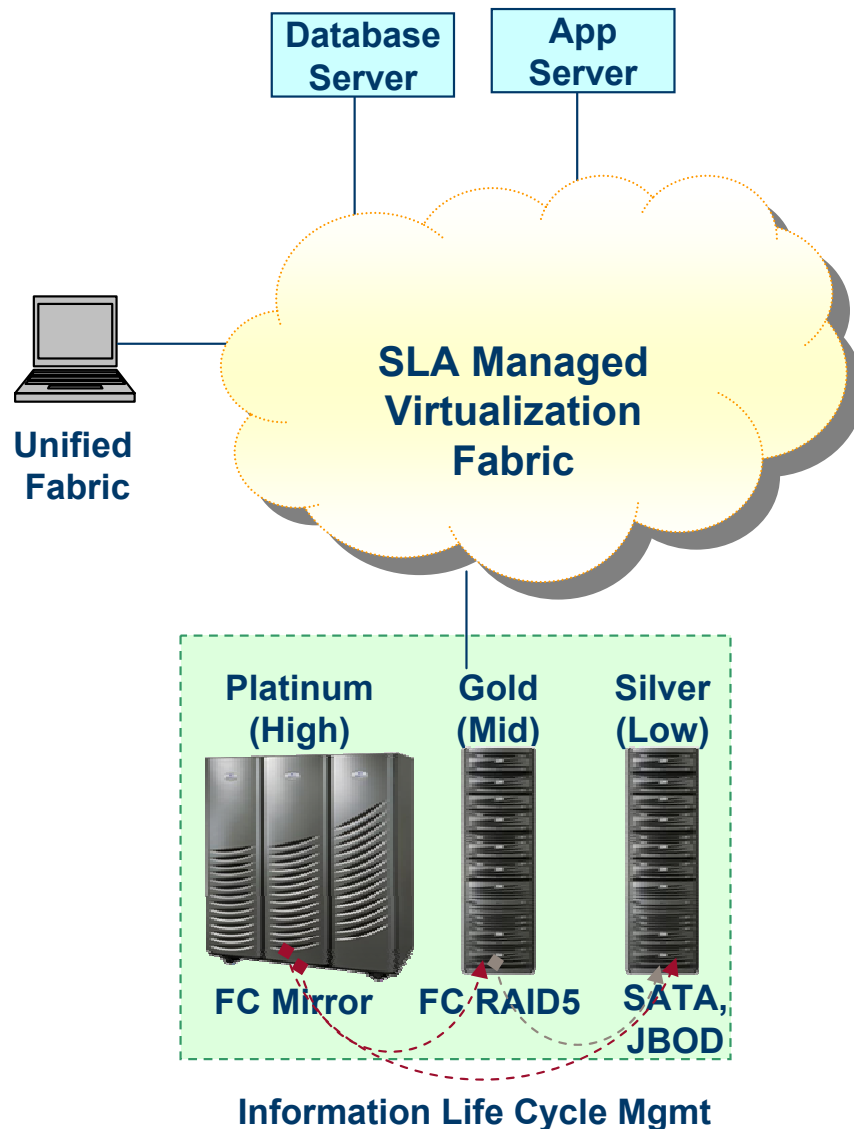
- OS → Fabric OS
- CPU → Machine
- Threads → Process
- Process → Distributed Application
- Memory → Distributed Memory
- Filesystem → Distributed FS
- Memory Backplane → High Speed Interconnect



Native OS virtualization will become the core building block

Service Oriented Fabric

Storage Lifecycle Management



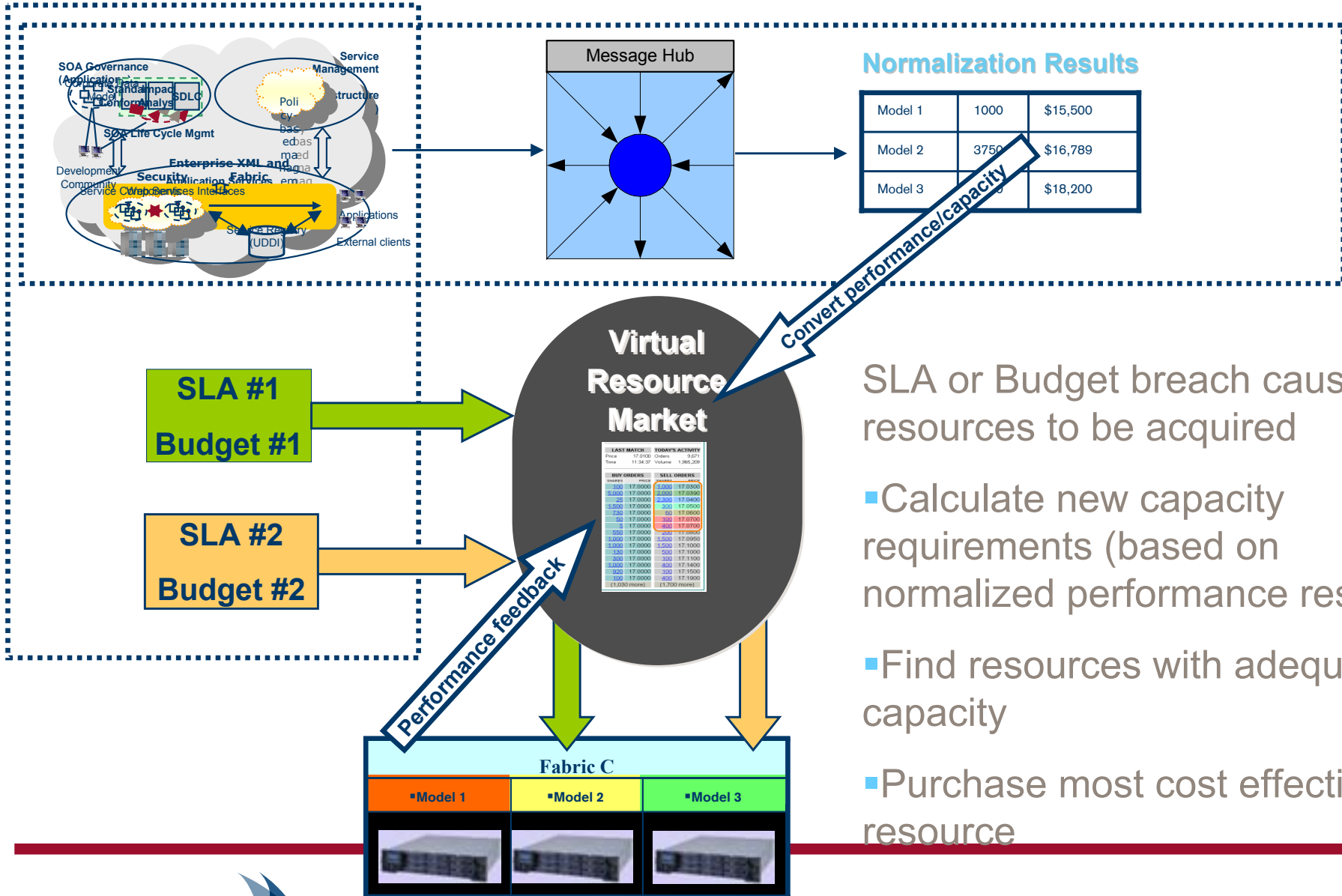
- **Manage Storage to meet changing Service Levels (e.g. resiliency, performance)**
- **Service Level Classification (Platinum, Gold, Silver)**
- **Automated Data Life Cycle Management – Not only HSM Vendor Neutral**
- **Unified Management Fabric**
- **Intelligent Switches**
- **Thinner Provisioning models for storage**

Information Life Cycle Mgmt

Automated management of business service level - best of breed storage

Service Oriented Fabric Capacity Management

SLAs and Budgets are the management for Service Oriented Fabric's VRM



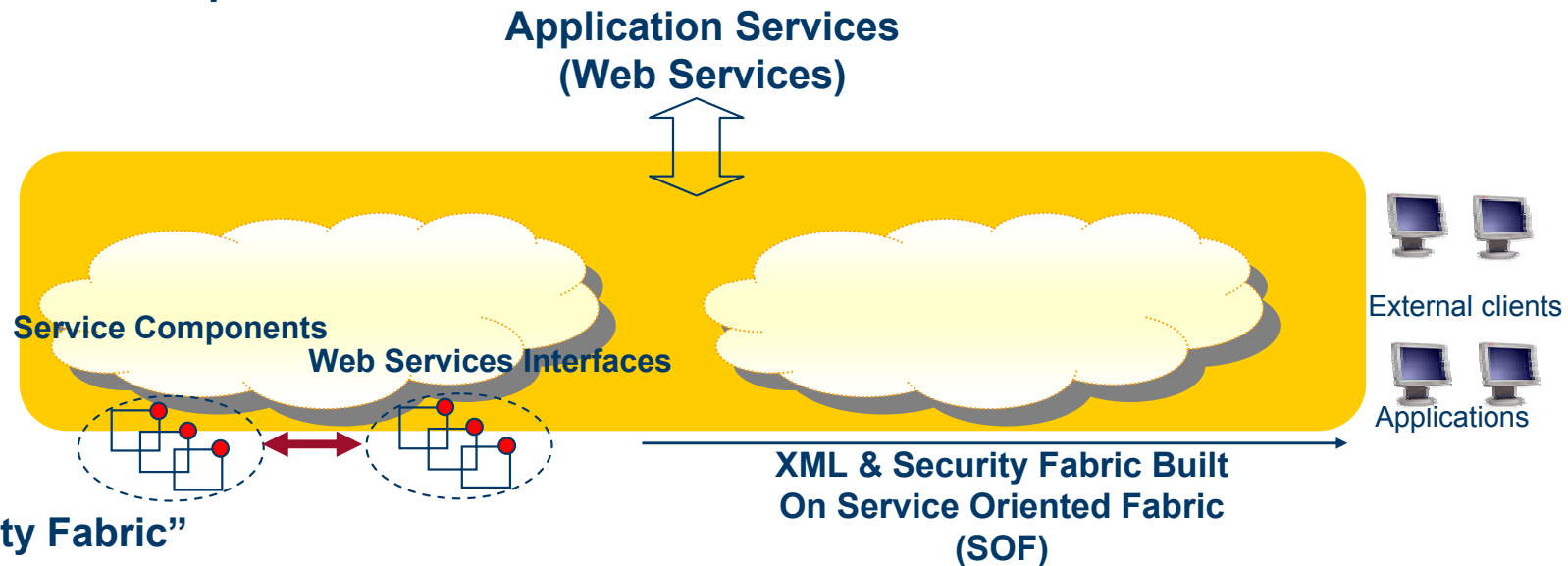
SLA or Budget breach causes new resources to be acquired

- Calculate new capacity requirements (based on normalized performance results)
- Find resources with adequate capacity
- Purchase most cost effective resource

Service Oriented Fabric

Enterprise XML and Security Fabric

- “XML Fabric”
- SOA governance and WSM must be coordinated to address the issues of managing the “XML fabric” that compose the SOA framework
- XML will become so engrained within the communications fabric that it will deteriorate mandatory service levels! Create unachievable service levels!
- Must be treated as an evolving ecosystem, thereby imbedding acceleration seamlessly within the new network platform



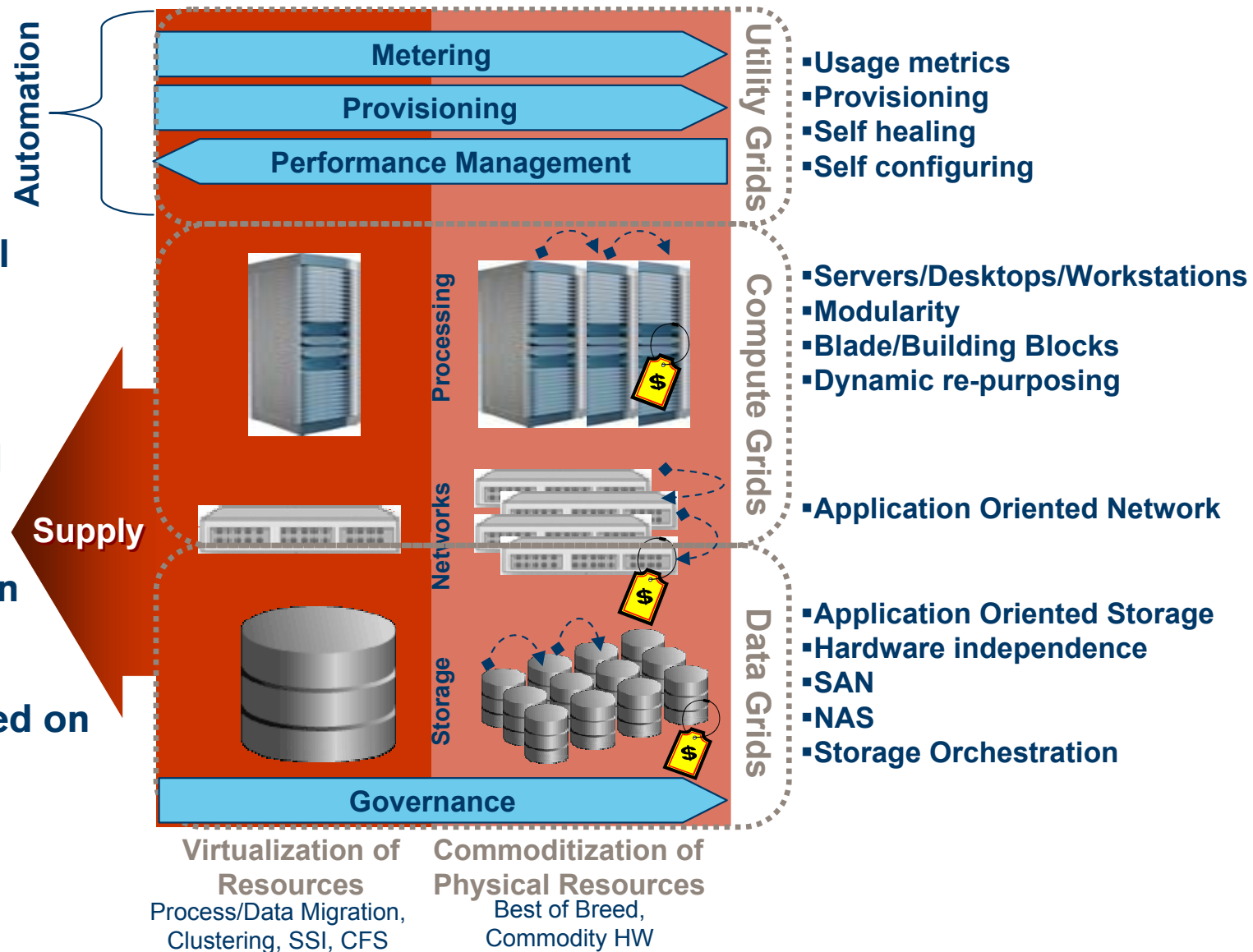
“Security Fabric”

- Security requirements must be treated holistically
 - Security management must be implemented and configured independently of application implementation
 - Security must be closely tied to contents of the XML fabric built on top of SOF
- Security must shift from the edge to embedded within the SOF
 - To work like Human immune systems

Service Oriented Fabric

Infrastructure Ecosystem To Simplify IT

- Solves the operational complexity
- Drives up agility
- Solves network based computing
- Breaks the squeeze on innovation
- Delivers capacity based on demand



Enables business that differentiates on interoperability and/or performance

Service Oriented Fabric: The Need For Dedicated Bandwidth & A New Way To Transmit Data

Service Oriented Fabric

Managed Bandwidth and New Data Transport

- Un-Managed bandwidth is obviously going to lead congestion, and just using priorities is going to be a potential management issue as one and all will state my application is the highest priority- so how do you virtualize everything in the datacenter (server, storage, network & datacenter) to create a Service Oriented Fabric?
 - We must also consider putting clients into the datacenter, and connecting them directly to the same fabric as the servers they communicate with (this will avoid hops, and reduce latency and could result in increased bandwidth for an application)
- Following the Clean Slate Mandate, everything's on the table - TCP/IP has existed for 30+ years- it was not meant for the datacenter, and developers are spending far too much time working out latency issues with TCP/IP as well as OS developers. Therefore we propose the concept of Data Highway Lanes (DHL) to describe a dedicated piece of bandwidth- this would not be a contracted piece of bandwidth, but rather a dedicated amount of bandwidth that is pre-allocated via a XML scheme describing the allocations of the DHLs; the amount of DHLs would be determined by the amount of physical bandwidth that could be carved up through allocations not by an algorithm in a piece of firmware or some one's predetermined notion of how many should DHLs should exist – we need a new to transfer data...

Service Oriented Fabric

Dedicated Bandwidth– EX: Not Completed XML

```
<some_name>
  <Application_Requirements>
    <1>
      ...
    <n>
  </Application_Requirements>
<Identity_Management_Requirements>
  <1>
    ...
  <n>
</Identity_Management_Requirements>
<Fabric_Requirements>
  <Physical_Logical = REQUIRED>
    /* HeartBeating */ <Logical = 0>
      <bandwidth = 10 B/S>
      <priority = 0>
    /* Server-Server */ <Logical = 1>
      <bandwidth = 1 GB/S>
      <priority -= 0>
    /* Server-Client */ <Logical = 2>
      <bandwidth = 1 GB/S>
      <priority = 1>
    /* Storage */ <Logical = 3>
      <bandwidth = 5.5 GB/S>
      <priority = 0>
    /* CFS */ <Logical = 4>
      <bandwidth = 2.5 GB/S>
      <priority = 0>
    /* Sniffing */ <Logical = 12>
      <bandwidth = 4GB/s>
      <priority = 0>
    /* Debug */ <Logical = 13>
      <bandwidth = 2 GBs>
      <priority = 1>
  </Fabric_Requirements>
</some_name>
```

Service Oriented Fabric

A New Way To Transport Data? - MEMORY

- Through the implementation of a Global Data Transport Layer (GDTL), it our intent to purpose to modify the boot-up options of the Linux kernel to allow for a pre-determined contiguous piece of memory be pre-allocated at kernel boot-up time whose use would not impact any kernel functions; rather, a userspace application would request an amount of memory, and get a pointer to memory. An application would have the ability to read, write and lock regions of memory.
- A DHL would used to allow data to transported from one application to another without the application being aware the transport was happening, and then the data is placed into the GDTL; it is further recommended that a sophisticated HCA could be developed in which interrupts could be sent to an application waiting on receiving data; thus instead of using socket, and application uses regions of memory to transfer data as well as using this same data space as part of his application instead of every application creating its own memory space.

Service Oriented Fabric

A New Way To Transport Data? - MEMORY

- This would not be the first time such a scheme has been purposed, and in the past there have several implementations; however, these implementations used spinlocks to sleep and wake-up to look for a change in data; it is the proposal of this Clean Slate presentation that either of the device driver or the HCA would have the intelligence to link to a DHL and send an event whenever new data was present or only when new data is present. Unlike the advocates of TCP/IP we do not suggest this model for every possible use case, but for use within a datacenter.
- Furthermore, attaching solid-state storage devices via PCI-Express Gen-2 or even the latest existing PCI-Express extensions, would allow the ability to access in some devices 1/2 TB of memory (tomorrow over a TB) could bring into the question the lossless fabric built on top of TCP/IP using a "PAUSE" to slow down data sending/receiving as opposed to a means to put data into a GTDL which then puts the burden on the decision of "stale" data on the application and not the switch and NIC causing "stale" data.

Service Oriented Fabric

A New Way To Transport Data? - MEMORY

- If you think this GDTL would not work... HP still has not removed the documents so you can still see them... Digital, later Compaq AlphaServer team created a memory channel that had rails, created a Global Address Space that also included a Distributed Lock Manager to lock regions of memory in a cluster... far superior to TCP/IP
 - Can still find the documentation even though what many called TRUE SOLUTION is no longer is for sale: http://h30097.www3.hp.com/docs/cluster_doc/cluster_50/TCR_MIG/TITLE.HTM
 - ↳ Take a look at Chapter Ten... it discusses locking and sending memory – more docs are still out there
 - ↳ If you interested in taking part in GDTL, contact Head.Bubba@credit-suisse.com
 - Here is some sample code on the next two slides...
 - ↳ If you want I have a entire sample piece of code that does sub-second failover using memory channel to lock a region of memory based on API released in Tru64 UNIX in April 1999 – also shows how to transmit data to another application (it is commented out, but did work) – if u email, I'll send it to you!

Service Oriented Fabric

Memory Channel Code – Transmit Address

```
long createTransmitAddress( memoryChannelAccess *pMemoryChannel ) {  
  
    int rc = 0;  
  
    if ( pMemoryChannel->pTransmitRegion ) return -96;  
  
    pMemoryChannel->pTransmitRegion = ( memoryChannelRegion * ) malloc( sizeof( memoryChannelRegion ) );  
  
    memset( pMemoryChannel->pTransmitRegion, 0, sizeof( memoryChannelRegion ) );  
  
    pMemoryChannel->pTransmitRegion->transmitDirection = IMC_TRANSMIT;  
  
    pMemoryChannel->pTransmitRegion->regionSharingMode = IMC_SHARED;  
  
    pMemoryChannel->pTransmitRegion->directionFlag = IMC_LOOPBACK;  
  
    rc = imc_asattach(pMemoryChannel->regionId, pMemoryChannel->pTransmitRegion->transmitDirection,  
                    pMemoryChannel->pTransmitRegion->regionSharingMode, pMemoryChannel->pTransmitRegion->directionFlag,  
                    &( caddr_t )( pMemoryChannel->pTransmitRegion->pAddress ) );  
  
    if ( rc ) {  
  
        imc_perror( "Failed to initialize transmit address for memory channel - imc_asattach::", rc );  
  
        return rc; }  
  
    pMemoryChannel->attachedRegion = 1;  
  
    return 0; }
```

Service Oriented Fabric

Sample Memory Channel – Receive Address

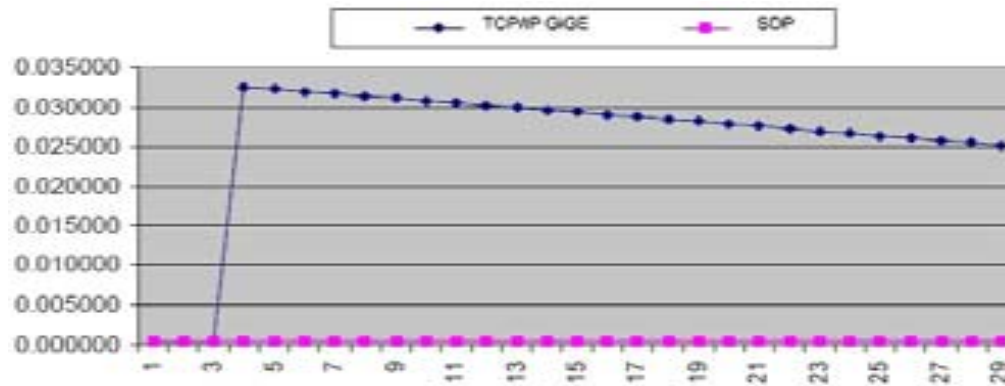
```
long createReceiveAddress( memoryChannelAccess *pMemoryChannel ) {  
  
    int rc = 0;  
  
    if ( pMemoryChannel->pReceiveRegion ) return -196;  
  
    pMemoryChannel->pReceiveRegion = ( memoryChannelRegion * ) malloc( sizeof( memoryChannelRegion ) );  
  
    memset( pMemoryChannel->pReceiveRegion, 0, sizeof( memoryChannelRegion ) );  
  
    pMemoryChannel->pReceiveRegion->transmitDirection = IMC_RECEIVE;  
  
    pMemoryChannel->pReceiveRegion->regionSharingMode = IMC_SHARED;  
  
    pMemoryChannel->pReceiveRegion->directionFlag = 0;  
  
    rc = imc_asattach( pMemoryChannel->regionId, pMemoryChannel->pReceiveRegion->transmitDirection,  
                     pMemoryChannel->pReceiveRegion->regionSharingMode, pMemoryChannel->pReceiveRegion->directionFlag,  
                     &( caddr_t )( pMemoryChannel->pReceiveRegion->pAddress ) );  
  
    if ( rc ) {  
  
        imc_perror( "Failed to initialize receive address for memory channel - imc_asattach::", rc );  
  
        return rc; }  
  
    pMemoryChannel->attachedRegion = 1;  
  
    return 0; }  
  
}
```

Service Oriented Fabric Conclusions

Service Oriented Fabric

Conclusions

- The desire is to have well defined, deterministic latency
 - Currently, applications send data via TCP, and we've seen high variance
 - TCP/IP Client Latency (GiGE) as compared to SDP on another fabric



- What exists today is not good enough for tomorrow – we need something better
 - The way things are done today does not make them right for tomorrow
 - Live application migration to any fabric (an option that should be considered)
 - Clock Synchronization for all hosts on a fabric
 - Ideally: Guaranteed Service Level Agreements (SLAs) for Bandwidth and Latency
 - More un-managed bandwidth is not the solution
 - wider roads invite more traffic... and ensuing congestion

Service Oriented Fabric

Conclusions - continued

- Increase in volume of data passed between applications have created the need for lower latency solutions
 - TCP/IP has been found to be problematic for some classes of applications & was designed for wide area communication
 - Gateway window size (mtu) and "connectedness" are indeterminate
- Virtualize Entire Data Center Creating A Service Oriented Fabric
 - Virtual Resource Market (VRM) managed by a QoS/SLA Manager
 - Migrations based on SLAs & Real Time Data Center Monitor interaction
 - ↳ Server
 - ↳ Fabric
 - ↳ Storage
 - ↳ Etc
 - Help Create The Global Data Transport Layer (GDTL)
 - ↳ Several have already joined, several considering – if you want to join the project contact both Head.Bubba@credit-suisse.com & Steve.Yatko@Credit-Suisse.com
- Open Fabrics Alliance (*Credit Suisse is a Board Member*)
 - Open Standards & Open Source Solutions

Q&A and Contact Info

Q & A

■ Contact Information

– “Head” Bubba

VP

Head of Systems and Network Fabric Research

IT Research & Development

Credit Suisse

11 Madison Avenue

New York, New York 10010

email: head.bubba@credit-suisse.com

head.bubba@ieee.org

■ *To Contact Head of IT Research and Development*

▪ *Or To Look Into Our R&D Out-Reach Partnership Programs with Universities and Vendors*

– Steve Yatko

Managing Director

Head of IT Research & Development

Credit Suisse

11 Madison Avenue

New York, New York 10010

email: steve.yatko@credit-suisse.com