Grid on Blades

Basil Smith
7/2/2005

What is the problem?

- Inefficient utilization of resources (MIPS, Memory, Storage, Bandwidth)
  - Fundamentally resources are being wasted due to wide and unpredictable dynamic range in workload burdens – static or pseudo static resource allocation schemes do not work.
  - Underutilized resources in:
    - In server farms
    - At client endpoints
- Constraints
  - Security: need to run most apps with glass house class security
  - Licenses: need to get as much bang for buck for each license (this puts very real constraints on utilization of highly fragmented resources)
  - Software conflicts – hosting of grid application on a shared OS raises serious problems with conflicts and compatibility – frequently does not work at all and testing for obscure interaction is prohibitive
  - Software compatibility - applications cannot be extensively rewritten, they tend to run in context of a specific OS, middleware, and cluster environment
  - Dependability: particularly with respect to data integrity

Some observations and context:

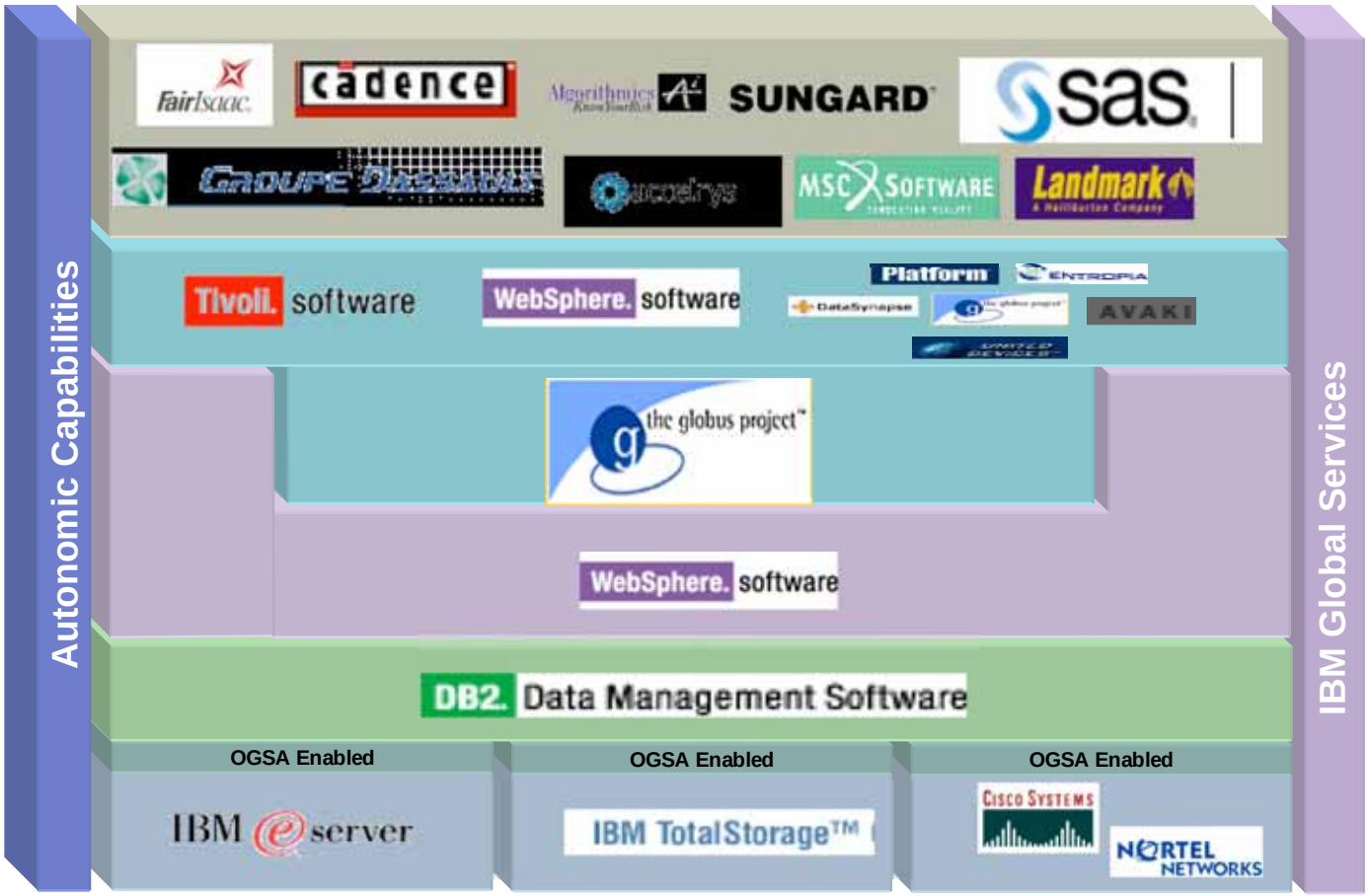
- Except for some very niche applications, trying to better utilize client endpoint resources is unproductive – why?
  - Security: no real solution exists, physical remains security essential part of picture.
  - Licenses: inefficient license utlitization wastes more than the value of the HW resources being retrieved.
  - Software conflicts: no efficient solution exists to assuring grid application will not conflict with client applications in shared host environments.
  - Software compatibility: OS/middleware/application stacks are mostly deployed using “clone” model, this would dictate reboot of client to grid clone image (or virtualization equivalent) – mostly this is an issue of switching from Windows client to Linux grid application.
  - Server hosting of clients (with thin display head) is likely a more effective means of addressing client resource waste.
  - Dependability: Dependability burden of using client HW on glass house core may be greater than payback – need for secure storage in anycase, and client storage is more inefficient than data center storage.
- Practicality dictates grid on/among scale out server farms

At the very bottom, what is the deployment model

- An application on a single node is deployed using “clone model”
  - Clone == boot disk image of OS/middleware/application instance, normally created from golden image, plus some customization
    - Virgin image – never been run no state beyond T0 image
      - Easily recreated from golden image
    - Dirty image – includes state changes from running image
      - May include extensive application state

Why Cloning – what's the application stack look like?

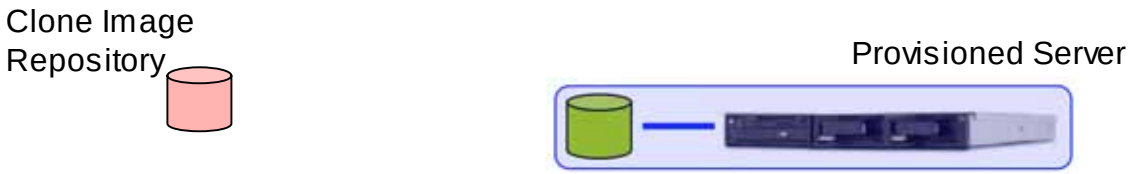
It looks like a bill board of stuff you need, and we will sell you ;-)



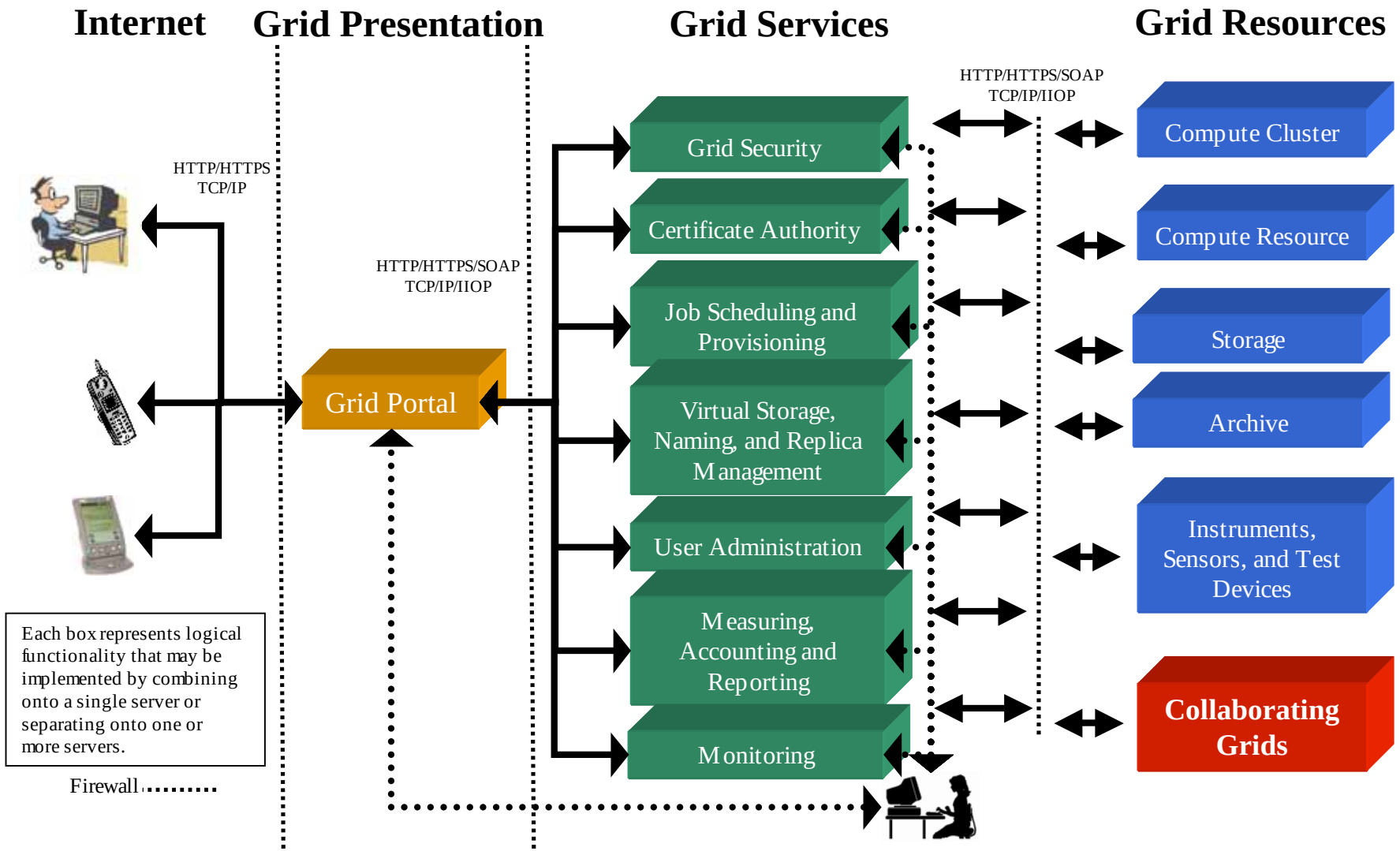
Build is tedious and release to "gold" is a lot of testing, somewhere in all of this you also might actually have to write some lines of code.

At the very bottom, retasking a server

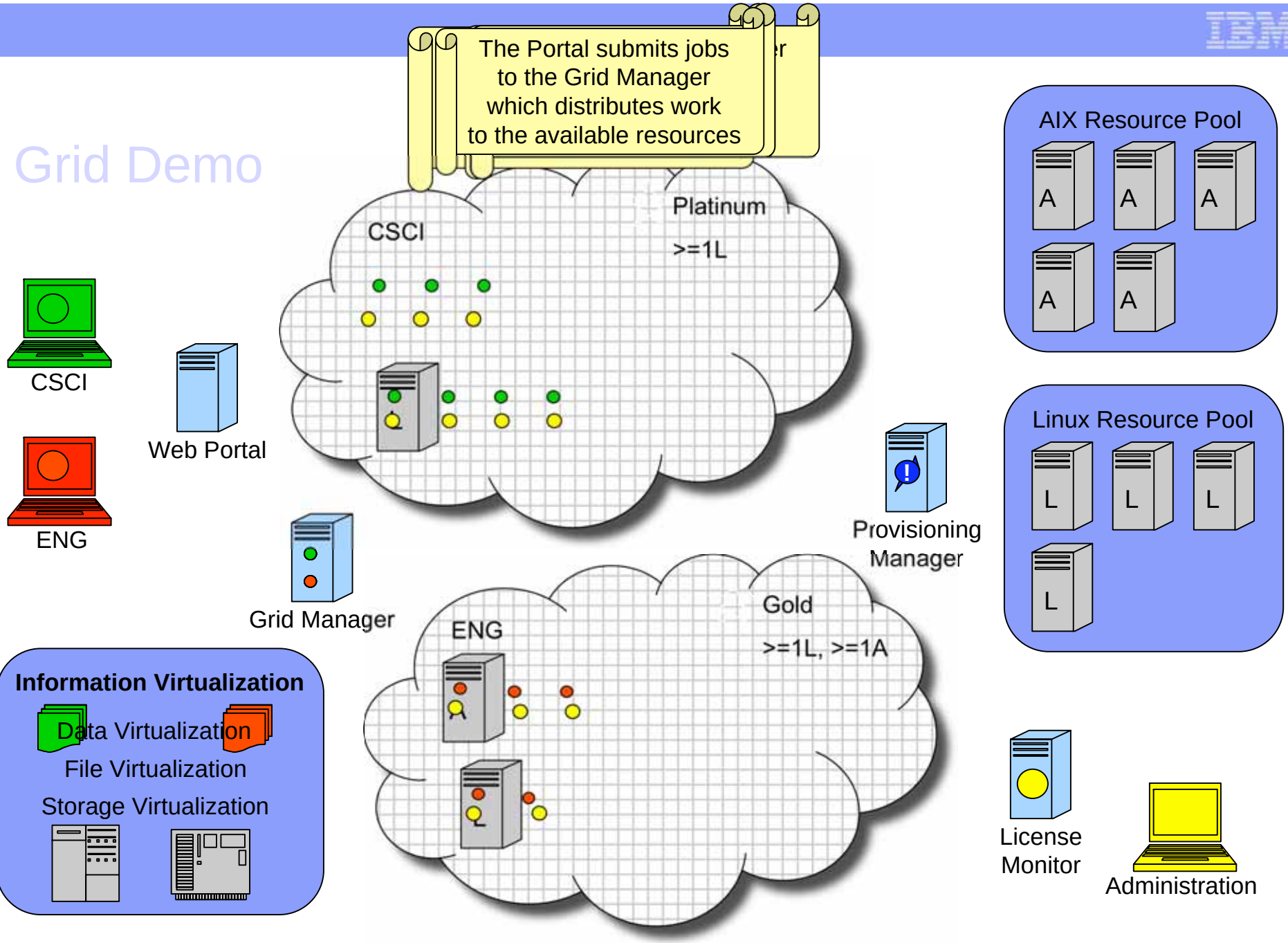
- To retask:
  - “Hibernate” an active server (force all state to disk – a dirty clone)
  - Turn server off
  - Disconnect dirty clone of that image from server
  - Connect new clone to server
  - Boot new image



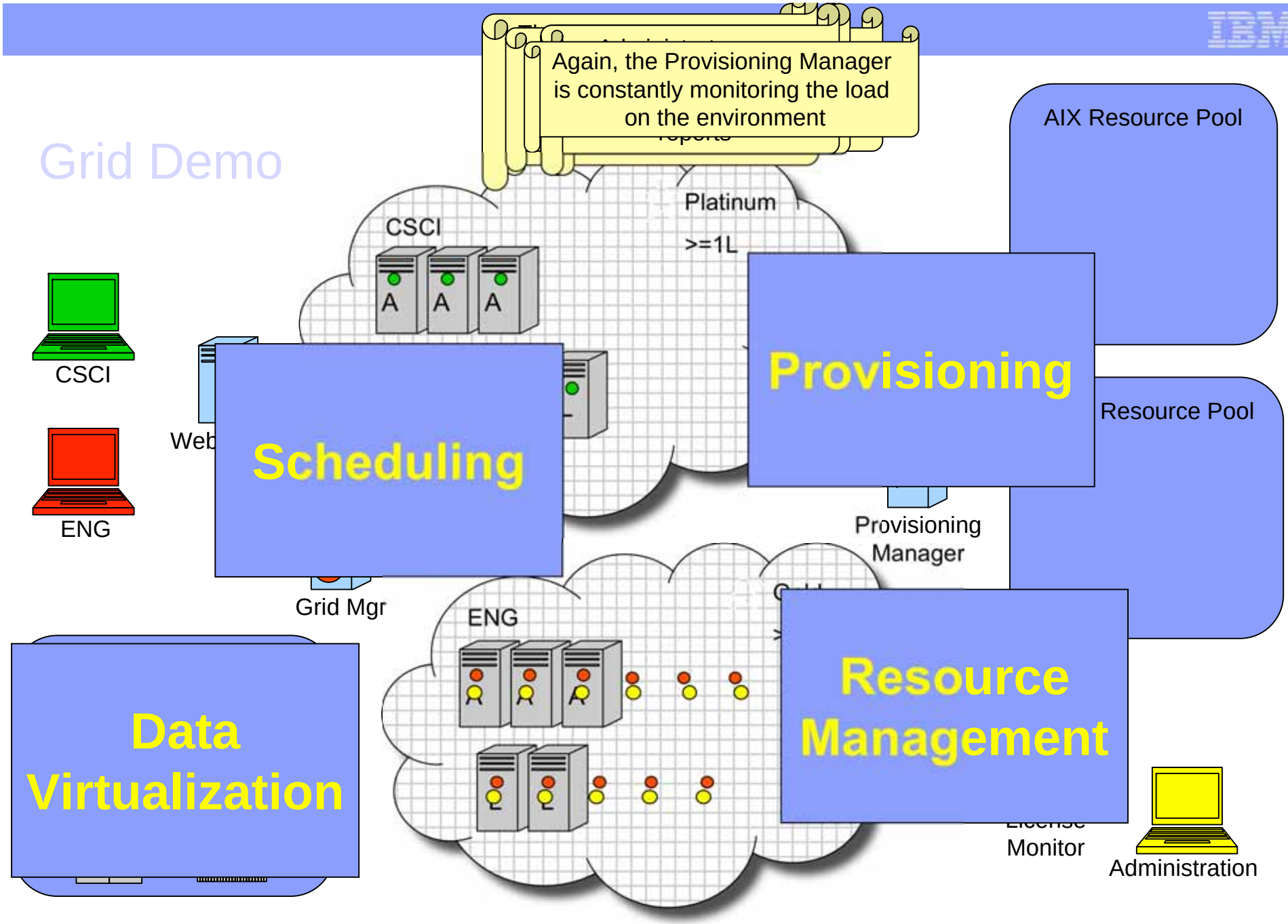
Grid Logical View



Grid Demo



Grid Demo



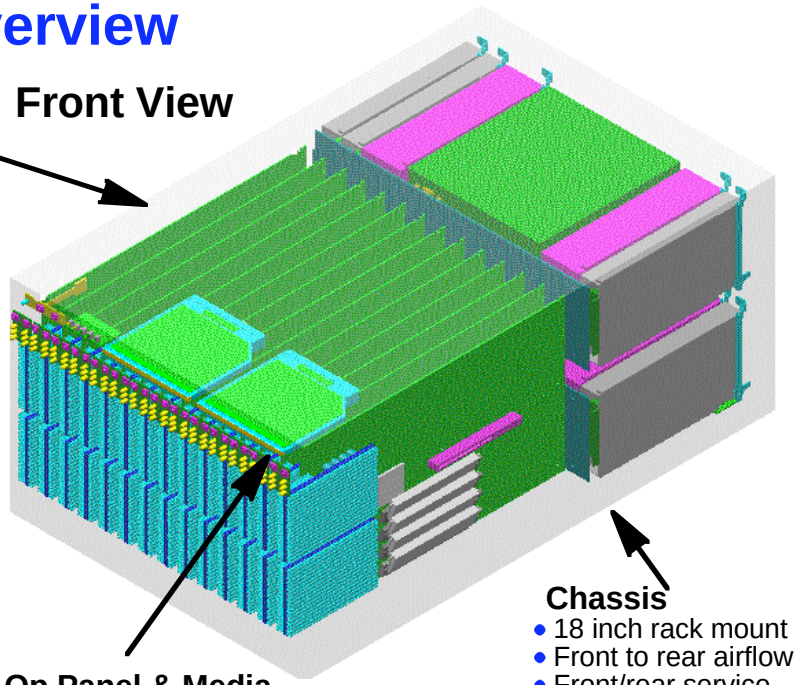
Again back to the bottom – what are these resources

eServer BladeCenter Overview

Processor Blades

- Hot swappable blades
- LEDs: Power, Alert, Info, Locate, Activity
- Buttons: Power, Reset, KVM Sel., Media Sel.
- USB, LightPath, Management, Video (HS)
- Processor Flexibility:
  - HS20 - 2-way XEON EM64T
    - 2GHz to 3.6 GHz, 800MHz FSB
    - 512MB to 8GB ECC memory
    - 2 Gb Ethernet + Opt. I/O feature card
    - Opt. to 2 SFF SCSI w/RAID0 or 1
  - HS40 - 4-way XEON MP
    - 2.0GHz to 3.0GHz, 400MHz FSB
    - 1GB to 16GB PC2100 ECC memory
    - 4 Gb Ethernet + two Opt. I/O feature card
    - Opt. to 2 SCSI disk via 'sidecar'
  - JS20 - 2-way PowerPC® 970
    - 2.2GHz, 800MHz memory
    - 512MB to 4GB ECC PC2700 memory
    - 2 Gb Ethernet + Opt. I/O feature card
    - Opt. to 2 IDE drives
- Optional - I/O Feature Cards:
  - Dual 2Gb Fibre Channel HBAs
  - Dual 1Gb Ethernet NICs (4 total)
  - 2Gb Myrinet cluster interface
  - Dual 1x InfiniBand HCAs
- Optional - dual SCSI disk 'sidecar'
  - 18.2, 36.4, 73.4, 146 or 300GB capacity
  - 10K RPM or 15K RPM speed
  - Built in mirroring, Hot swap
  - Two I/O Feature Card sockets
- Optional - dual adapter slot PCI-X 'sidecar'

Front View



Op Panel & Media

- Chassis level LEDs-
  - Power, Alert, Info,
  - Chassis 'Locate' indicator
- USB Port
- Removable storage media
  - CD & floppy disk

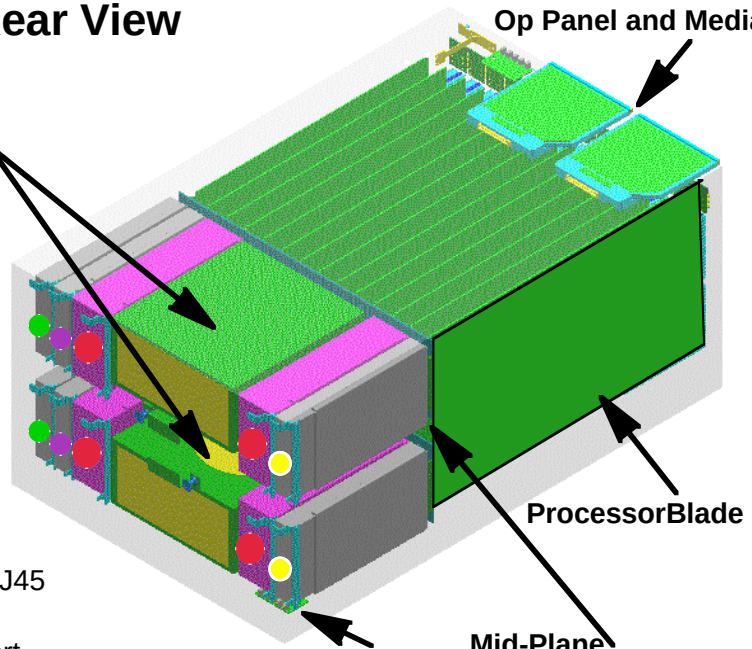
Chassis

- 18 inch rack mount
- Front to rear airflow
- Front/rear service
- Rear cabling

- "Enterprise" Rack
  - 14 CPU Blades
  - 7U high, 28" deep

- "Telco" Rack
  - 8 CPU Blades
  - 8U high, 20" deep
  - DC or AC pwr
  - NEBS ready

Again back to the bottom – what are these resources



Blower Module (2X)
- Hot Swap, Redundant
- 300 CFM, speed controlled

Optional Switch Module (0, 1, or 2)
- Hot Swap, Optional Redundancy
- Input: 14 blades + 2 MM (1-3Gb + 100Mb)
  - Ethernet - (same options as below)
  - Fibre Channel - Uplink: 1/2 Gb FC SFP
    IBM SAN Switch, Brocade SAN Switch
  - OPM - Direct optical link to each blade's port
  - InfiniBand - Uplink: 12/4x IB (40Gbps total)

Ethernet Switch Module (1 or 2)
- Hot Swap, Optional Redundancy
- Input: 14 blades + 2 MM (Gb + 100Mb)
  - IBM Layer 2 Ethernet Switch
  - Nortel Networks L2/3, L4-7, SFP or RJ45
  - Cisco Layer 2+ Ethernet Switch
  - CPM - Direct RJ45 to each blade's port

Power Module (2 or 4)
- 200-240V AC (worldwide volt./freq.)
- Hot Swap, Redundant (Opt.)

Mgmt Module (MM) (1 or 2)
- Chassis management control point
- KVM Switches (Local and Remote)
- Hot Swap, Redundant (Opt.)

Again back to the bottom – what are these resources

Processor Blade (Dual Xeon)

Low level management to enable grid

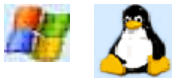
IBM Director

Server, Storage & Network Provisioning Tasks

Servers & Adapter Configuration

Fibre Switch Configuration

OS & Image Clone & Deployment



Storage Configuration

xSeries BladeCenter

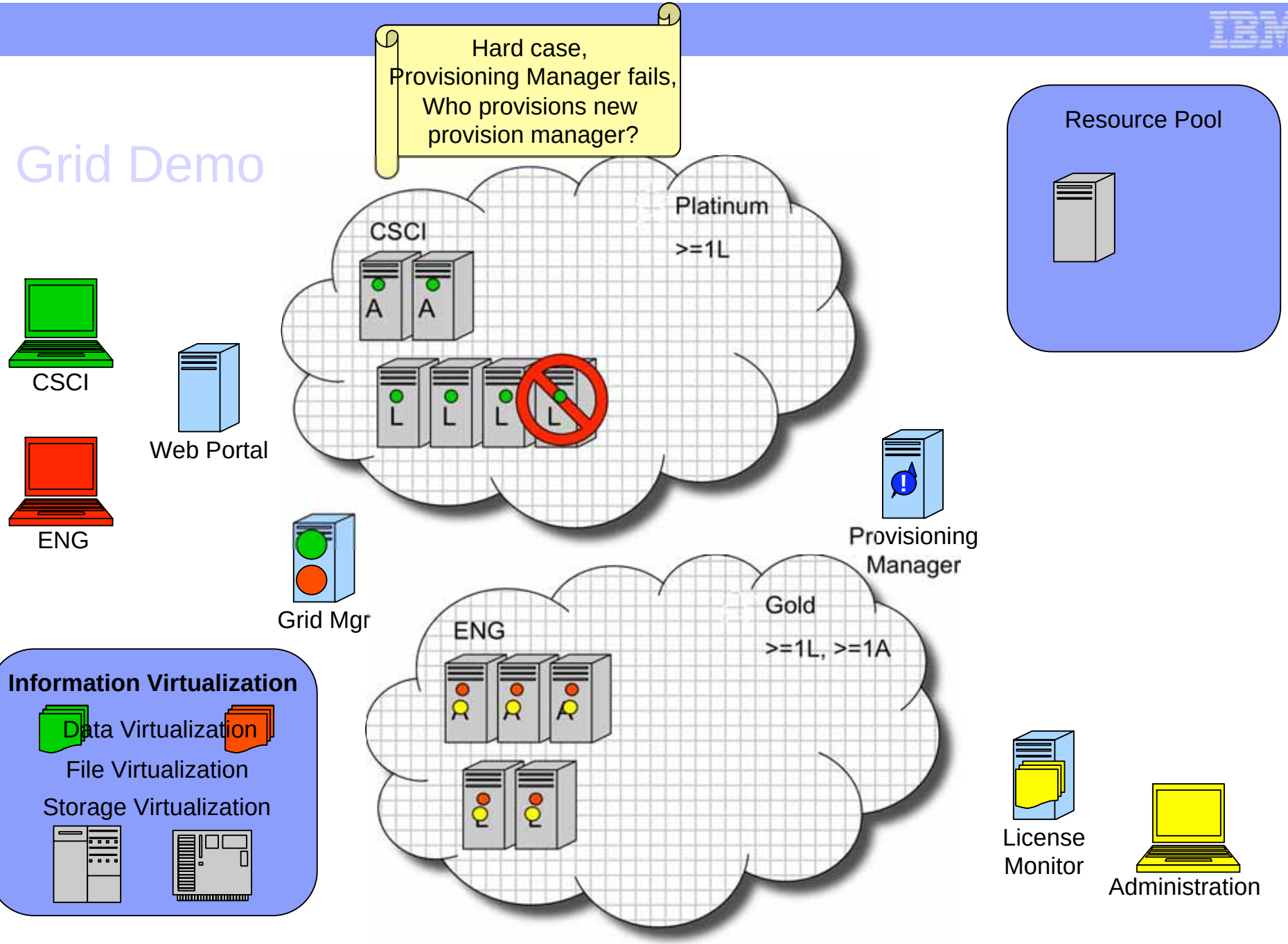
- Qlogic
- Brocade

- FAStT

Finally, the dependability challenge

- Break the problem down to known solutions
  - Classic cluster recovery for failed node in application
  - Reprovisioning of spare node to replace capacity
    - Is this with a virgin copy, checkpointed copy, or by just attaching failed image to another server and restarting
  - File and disk dependability and integrity management is critical, ultimately protecting against loss of state
    - RAID storage subsystems
    - Replicas and checkpoints (point in time copies)
    - Geographic replication (for disaster recovery)

Grid Demo

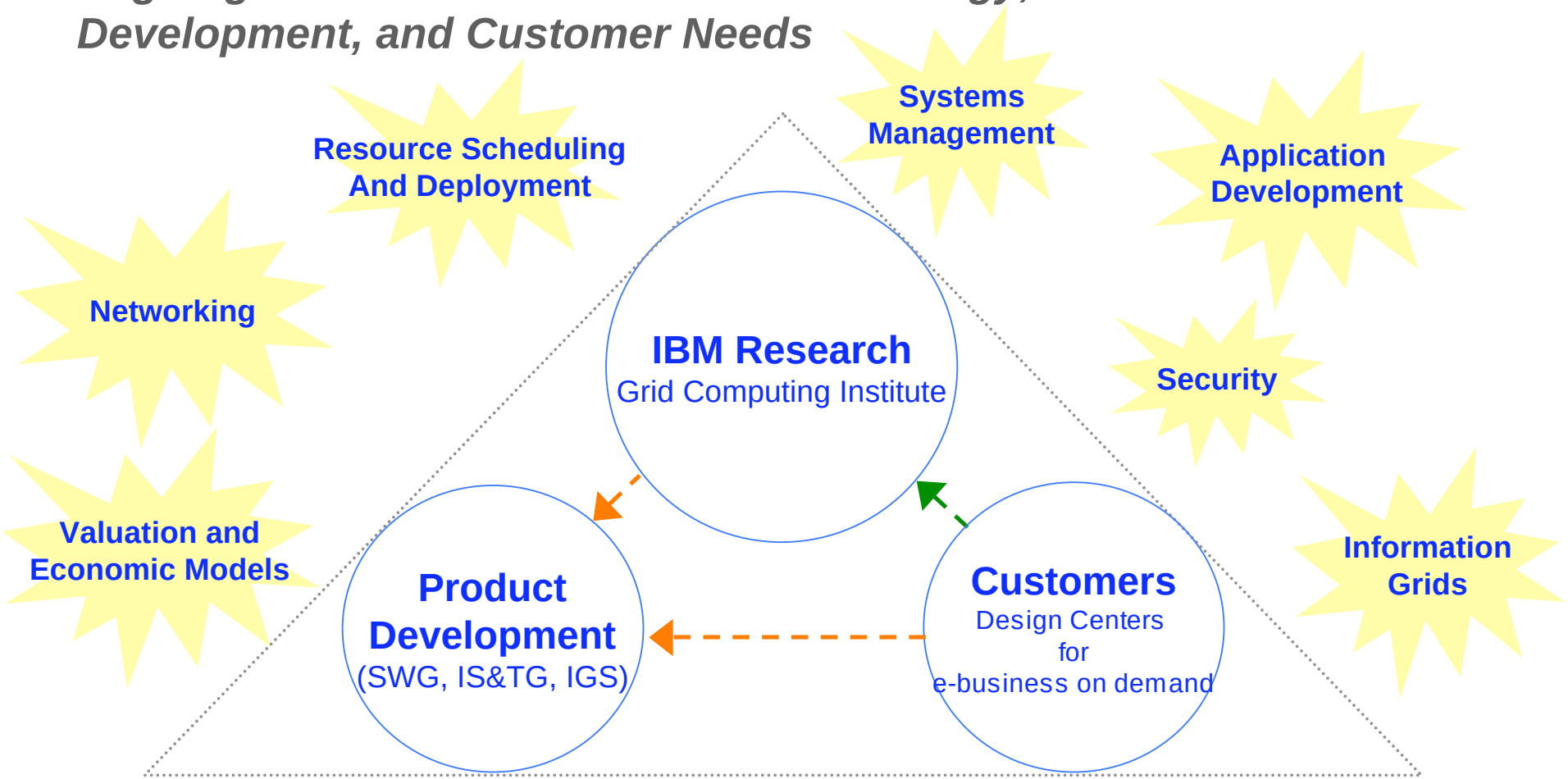


The dependability challenge

- Options / candidates for availability manager
- What grid services need to be availability aware
  - Lots of problems
    - Who recovers lost licenses
    - Strategy for recoverying basic grid services.
    - Break the problem down to known solutions
    - Who keeps compatibility matrix
    - Role of virtualization
    - Whats disaster recovery procedure for storage subsystem failure

Grid Computing Institute

Aligning IBM Research with the Grid Strategy, Product Development, and Customer Needs



Discussion: