Intel Data Grid

Gregory.Touretsky AT intel.com

Legal Notices

This presentation is for informational purposes only. INTEL MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

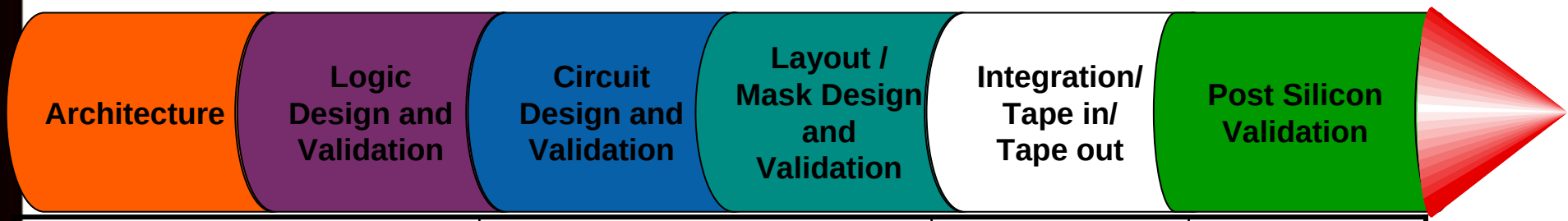
BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, Dialogic, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, IPLink, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, VTune, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Topics

- Intel Design Environment
- NFS-based storage grid - oxymoron or reality?
- Storage ATM concept and implementation

The Design Process Drives Computing

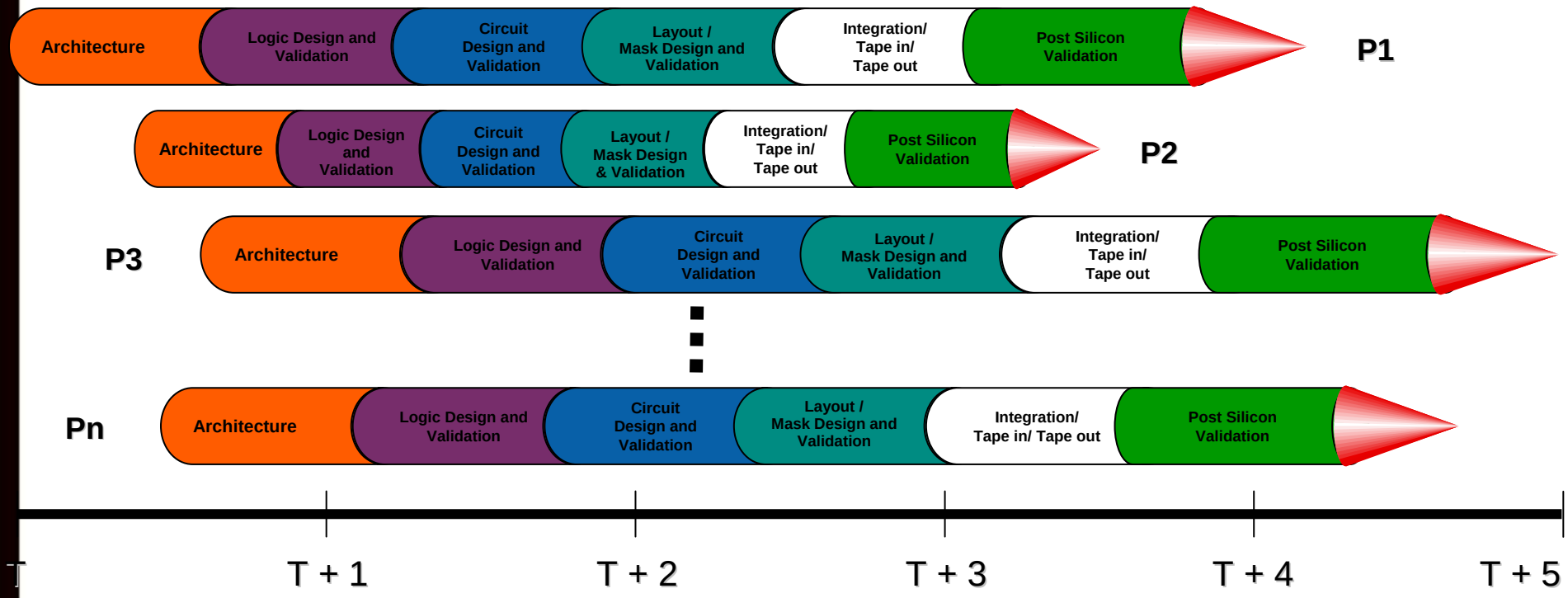


• Logic verification • Millions of small jobs • Primarily CPU-bound	• Interactive sessions (high end graphics) • Large datasets for schematics and layout DB's • Heavy I/O requirements	• Very high-end system configurations • 64Bit addressing • Multi-week job runs	• Lab /Test floor • Huge test databases • Special systems & software

70% of compute capacity

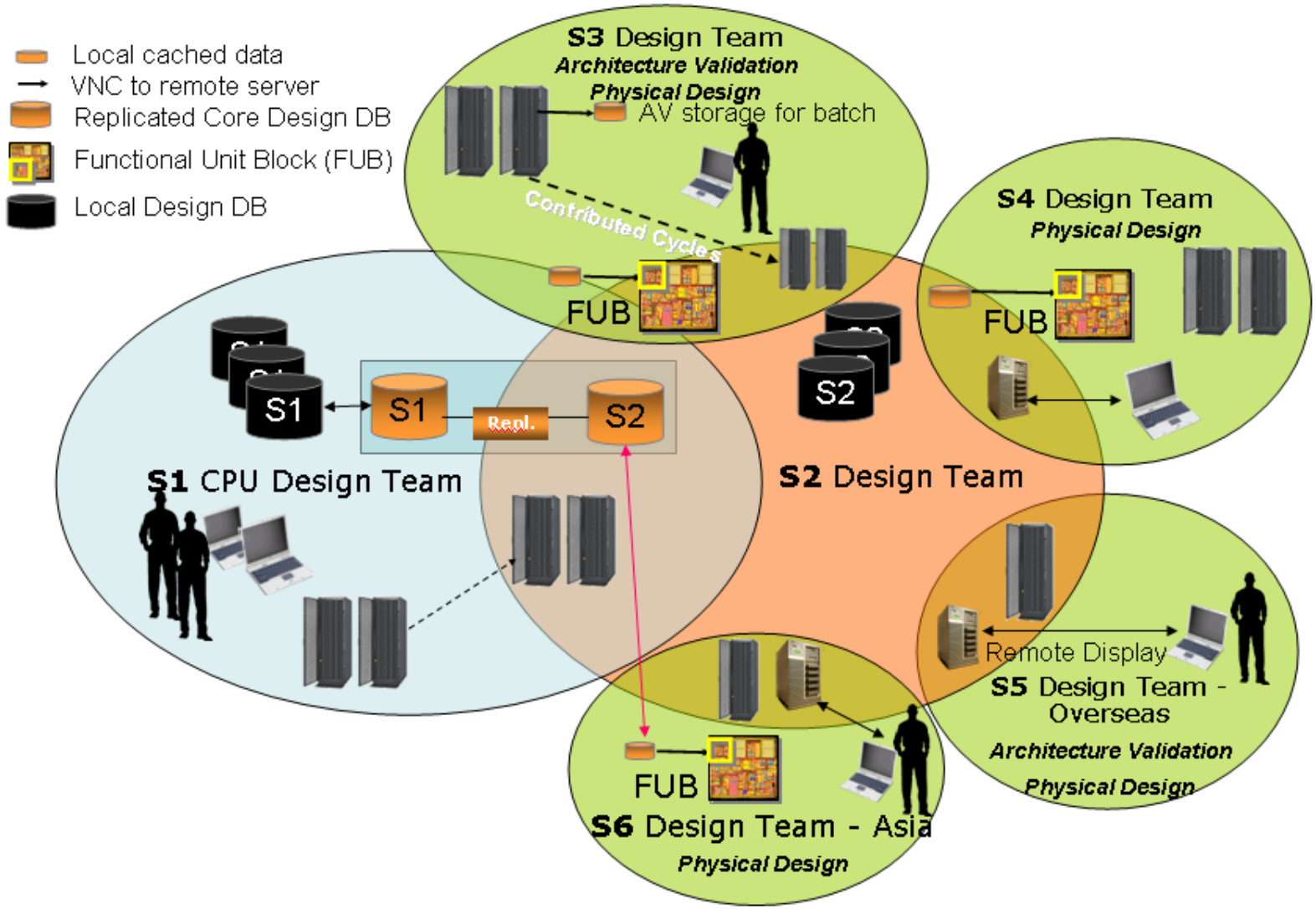
30% of compute capacity

Multiple Projects Run in Parallel

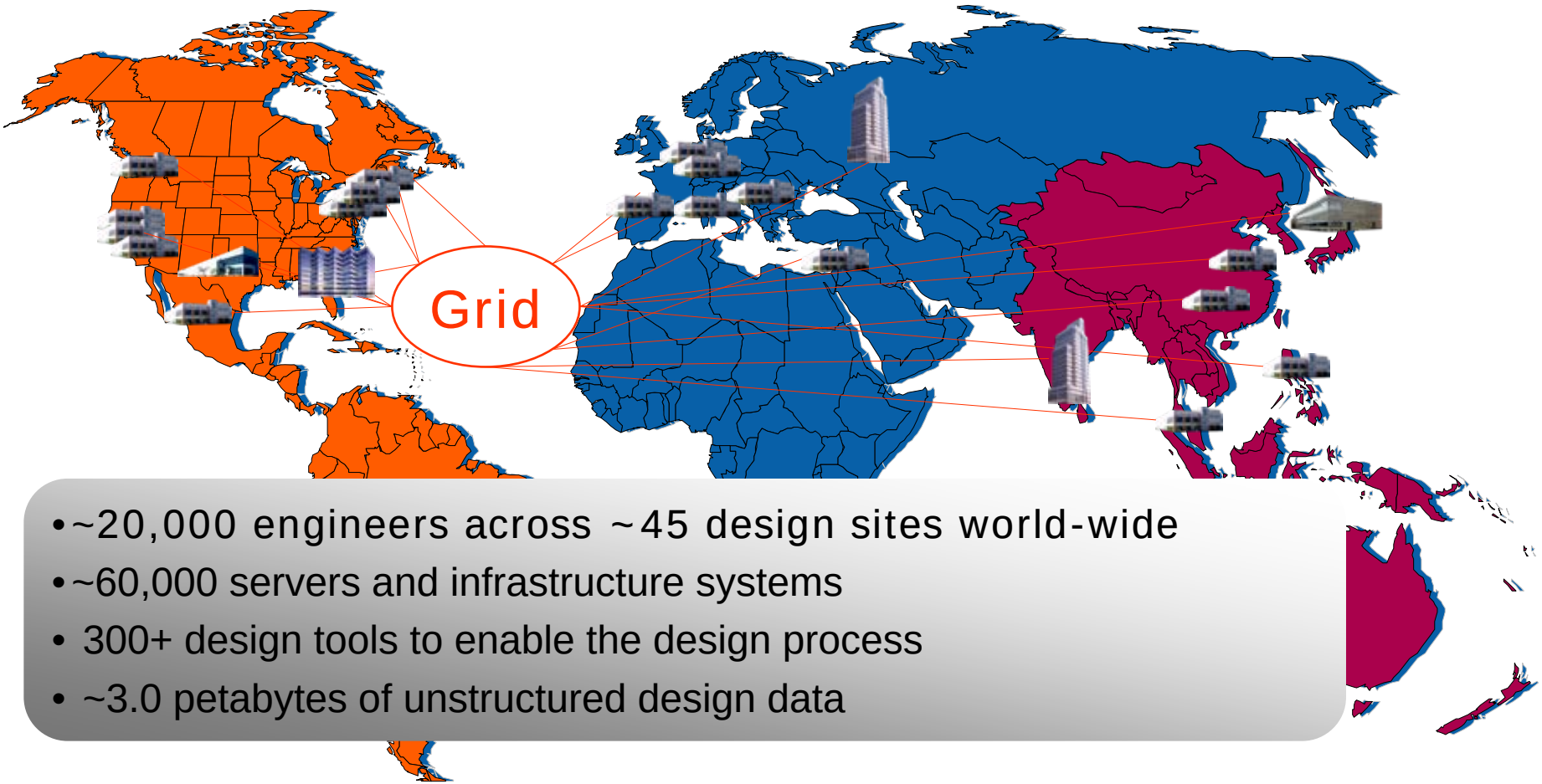


At any given time, we have a variety of projects running in parallel at different phases to develop a variety of products

Projects Span Geographical Borders



Intel Grid



- ~20,000 engineers across ~45 design sites world-wide
- ~60,000 servers and infrastructure systems
- 300+ design tools to enable the design process
- ~3.0 petabytes of unstructured design data

Selected OGSA Data and Resource Sharing Services - Requirements

- Global Name Space
  - Access entities transparently, subject to security constraints, without regard to location or replication
- Site autonomy
  - Access resources across sites while respecting local control and policy
- Policy specification & management
  - Who / Where / How fast / For how long / etc
- Data storage
  - Provisioning / Quotas / Encryption / Persistency
- Data access
  - Easy and efficient access to various types of data through uniform set of interfaces
- Data transfer
  - Point-to-point, multiple endpoints, …
- Data location management
- Authentication and Authorization

Reference:
http://www.ggf.org/documents/GFD.14.pdf

Global Name Space

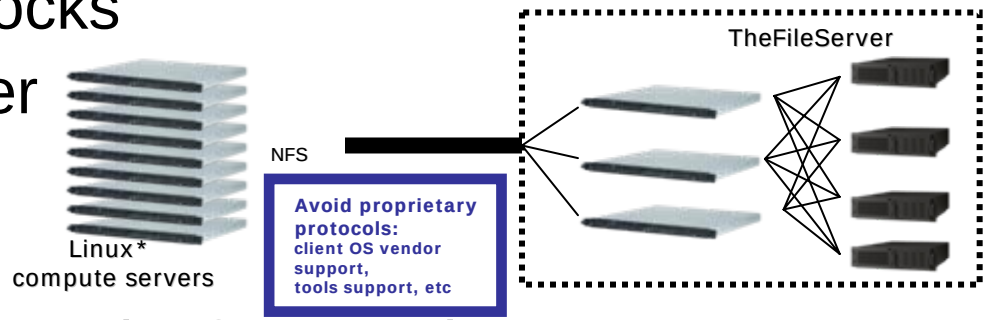
- Goals:
  - Decouple data path from physical location
  - Simplify data replication across sites
- How:
  - Chunk-based storage allocation
  - Automounter-based hierarchical name space
    - Project-based vs file server-based
    - /nfs/<site>/proj/<projName>/<mount>
  - Centrally-managed list of top-level “junctions”
- Gaps:
  - Lack of real “global” access solution – consider WAFS technologies and/or NFSv4 as potential direction

Data replication

- Goal:
  - Provide reliable, low latency and high throughput access to large amounts of data for globally distributed compute resources
- How:
  - Open-source and proprietary tools with in-house improvements
- Gaps:
  - No single global management and monitoring console – initiated in-house development
  - Access control is not complete
  - No easy way to reduce replicated data set to the absolute minimum (smart caching) – waste of disk space and WAN
    - WAFS may help – if ever becomes a reality

Storage performance scalability

- Goal: scale file server performance beyond the limits of traditional file servers. Appear as a single virtual file server to clients. Leverage Intel® architecture hardware as NFS cluster building blocks
- How: scale-out NFS cluster



- Gaps:
  - Emerging solutions, still lack of enterprise-level storage capabilities and support
  - Complexity
  - Lack of focus on design requirements (vs typical HPC requirements)

Control storage operations cost

- Goal:
  - Reduce amount of resources required for daily storage operation through automation
  - Avoid single-vendor lock-in for better cost control
  - Improve data quality for decision making
    - Purchasing, recycling, …

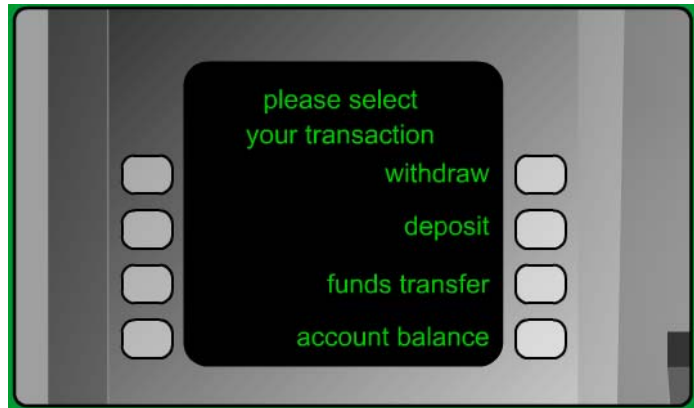
Storage ATM

(the analogy is courtesy of Dave Hitz @ netapp.com)

- Let customers to manage data themselves, without any help from the IT storage administrator
- Wanted: automatic, policy-based single point of control for all storage-related operations
  - Capable of managing 100s of file servers, PBs of data, 1000s of users at 10s geographical locations

Storage ATM, part 2

- Insert your card, enter PIN
- Select your transaction
  - Create/Delete/Rename/Resize storage area
  - Extend allocation period
  - Account balance – for you, project, site, server, …
  - Replicate directory
  - Change permissions
  - Update policies
  - Archive and Recycle
  - …



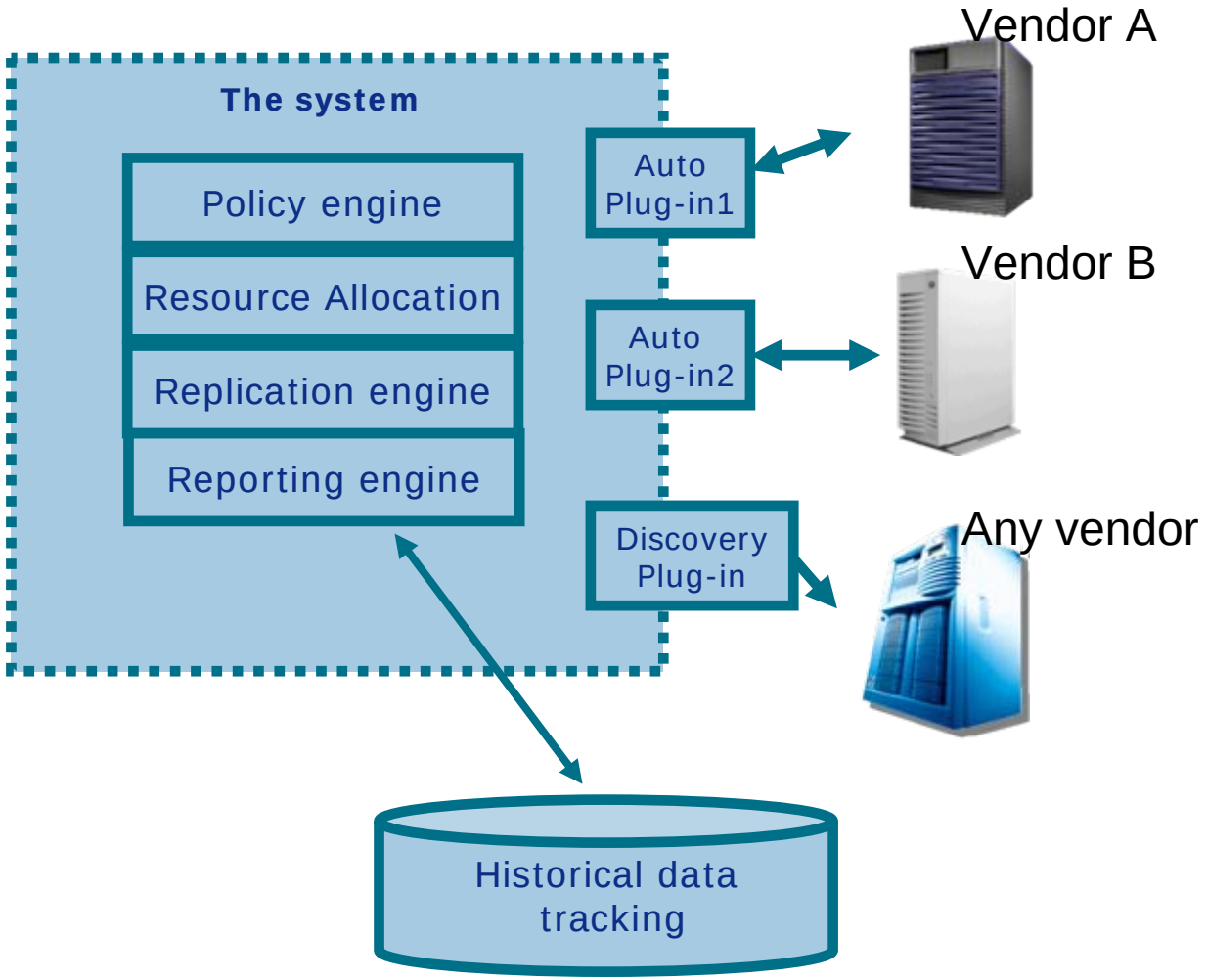
Inside "the ATM"

"ATM" operator:
- Defines Project and Project Resources

Project admin:
- Defines project members policy, project profile
- Manages permissions

End User:
- Disk allocation / reclaim
- Data replication
- Monitoring

Project manager:
- Global allocation reports
- Purchasing decisions



Policies

- A named recursive piece of logic
  - Storage Area policies
  - Partition policies
  - File server policies
- A triplet of Schedule, Condition and set of Actions
  - Condition defines an unwanted state - “bad situation”

```
SizeGB > 10 && Utilization < 0.5
```

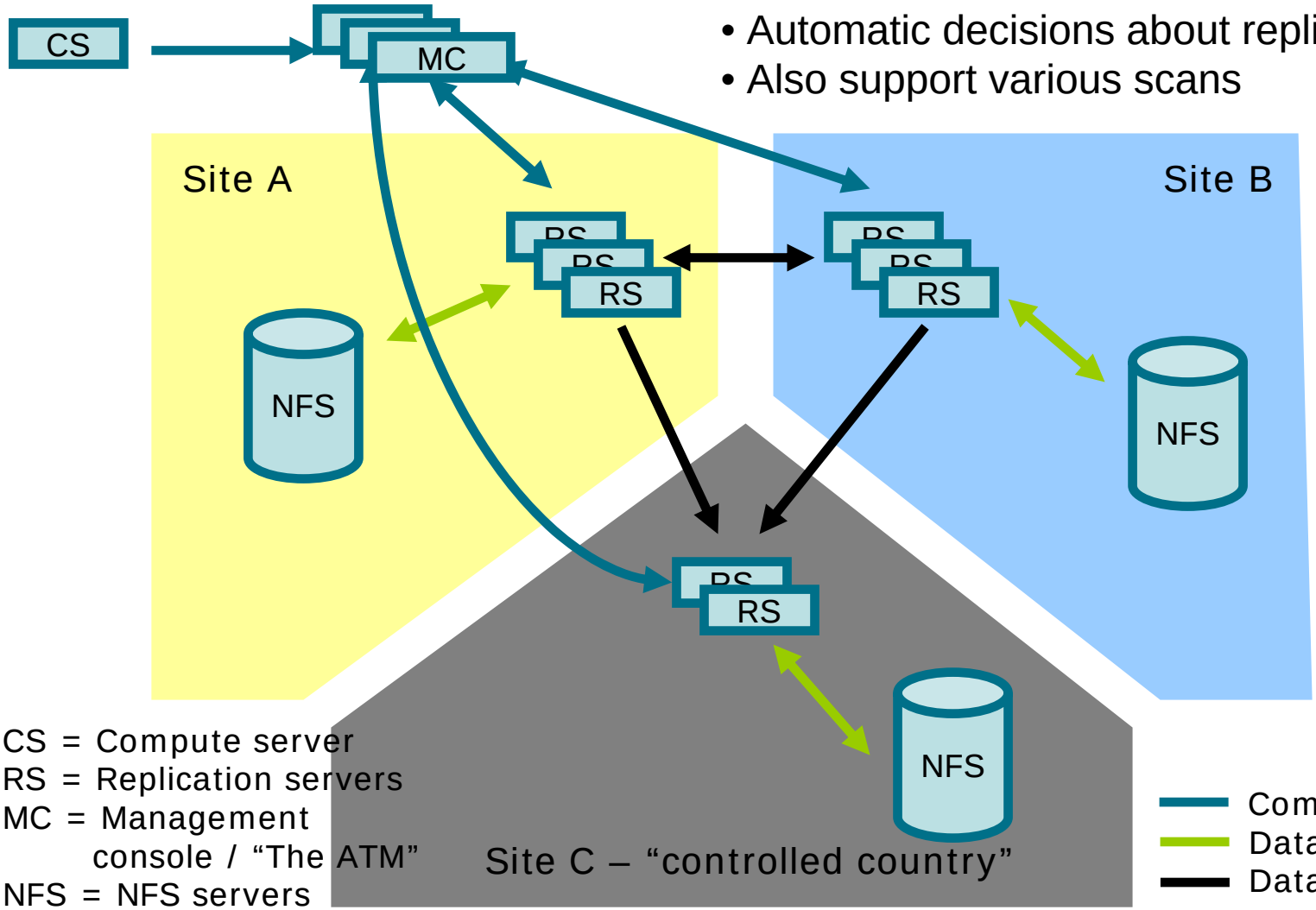
  - Actions try to fix it (mail/resize/delete/execute)
  - Schedule defines how often all this is done

Resource Allocation

- Hierarchical structure
  - Business group / Project / Sub-project / … / User
- Permissions
  - users – creating areas
  - admins – approving/creating areas for others
- Limitations for disk allocation
  - number of areas
  - maximum size of each area
  - maximum size of all areas
  - maximum allocation time
- Profile
  - can be defined on any level
  - defines defaults for:
    - file system permissions
    - allocation expiration options
    - notifications policies
    - requests approval definitions

Data Replication Model

- Automatic decisions about replication topology
- Also support various scans



Reporting

- Data aging
- Allocation
- Usage vs allocated vs forecast, trends
- Integration with the Data Warehouse
- Data transfers / WAN utilization
- …

"Storage ATM" – what's next?

- File servers performance monitoring
- Further integration with design flows
- Dynamic WAN topology definition for replication
- Thin provisioning – automatic decisions on over subscription rates
- Tiered storage support
- Template-based data pre-provisioning
- *Fully automate chip design… and make coffee!*

How do you address similar issues?

A question to both vendors and customers…

Contact: Gregory.Touretsky AT intel.com