



OppenheimerFunds

**How SOA and BPM are being used to improve
Operational Efficiency in the Mutual Fund
Industry**

September 8, 2008



- #3 largest provider of broker sold investment products
- ~ \$230 billion in Assets Under Management (AUM)
- Main offices in New York and Denver
- 2600 employees



The impetus for change at OppenheimerFunds

Business:

Customer servicing and our back office were not scaling in an environment of increased business velocity

- The cost to service each additional account was getting less and less attractive
- Both internal and external competitive pressure to service without manual touches (automation)

Technology:

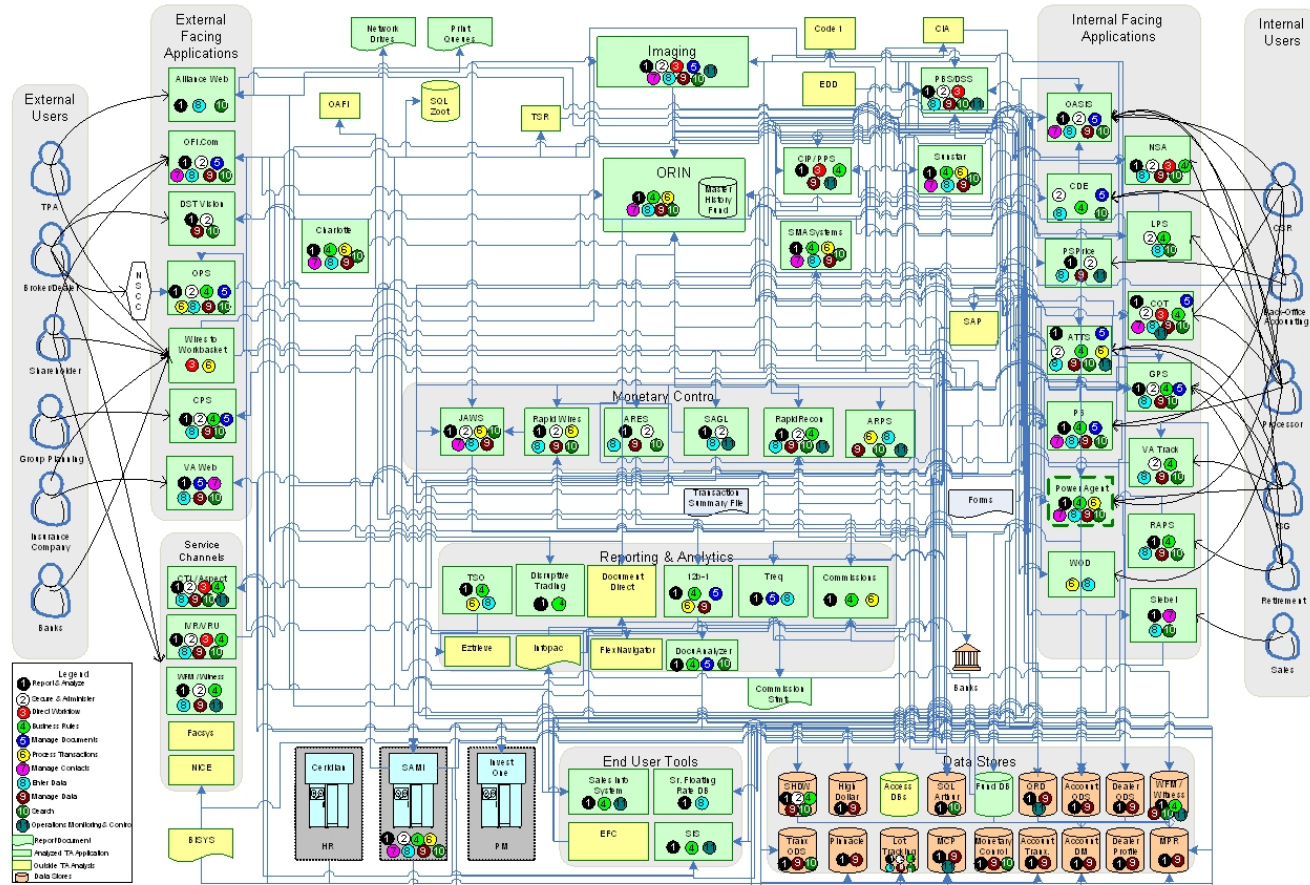
Years of neglected investment and patch work solutions left us with an accidental architecture and no clear way out of the situation



2006: Our Starting Point - It all made sense to someone?

Attributes of our Current State Architecture

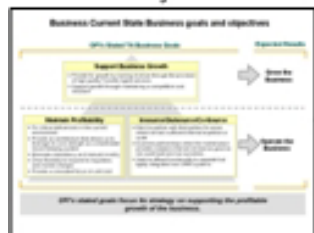
- Functional Redundancy in Applications
- Organizational application silos and limited cross-enterprise integration
- Inflexible System Integration
- Redundant and manual processes and technology
- Insufficient and redundant data
- Lack of flexibility and inherent scalability



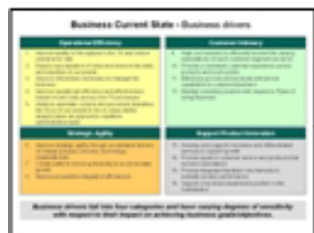
Strategic evolution → Lead with the business and rally around SOA



Goals & Objectives



Business Drivers



Value Chain





Strategic evolution → Lead with the business and rally around SOA

Goals & Objectives



Business Drivers



Value Chain



Work Streams



Conceptual Arch



Integration Arch





Strategic evolution → Lead with the business and rally around SOA

Goals & Objectives



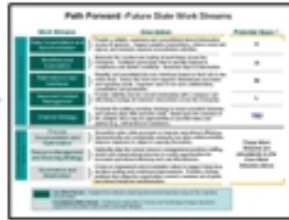
Business Drivers



Value Chain



Work Streams



Conceptual Arch



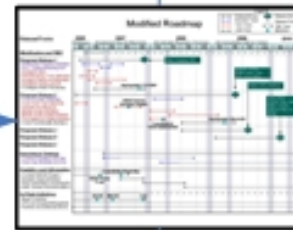
Integration Arch



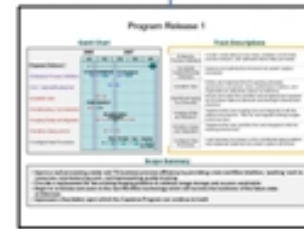
Phased Arch



Roadmap



Project Plans



SOA Blueprint



Service Context



Our technical strategy: Building on our core legacy environment



- Transfer Agency Application environment

2006 → • Legacy mainframe responsible for recordkeeping, customer processing rules, validation and reporting. PowerBuilder processing apps sit on top

2009 → • Legacy mainframe responsible for core recordkeeping with some customer processing rules. Major functions and information exposed via Services
↓

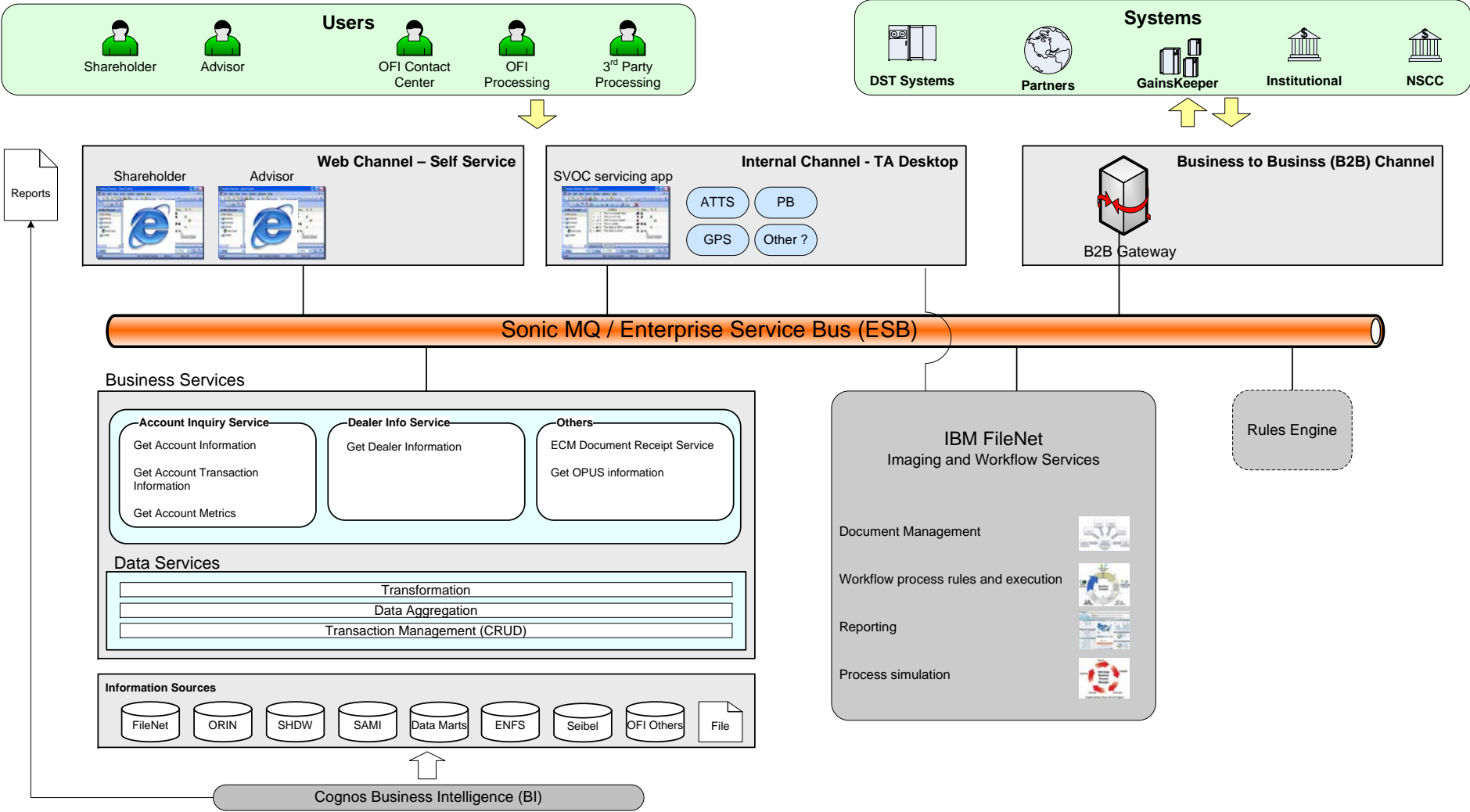
- Most value add functionality “offloaded” from mainframe onto external Services. *(Can always look for buy options when adding new functionality)*

- A new “efficient” UI sits in front a unified Customer view and Workflow driven processes.

- A focus on Shareholder servicing Workflows will mean encapsulating and extracting functionality currently held in legacy systems into Services exposed on the ESB *(More BPM/ESB integration)*

Win! • Shareholder Services are automatic candidates for reuse across other channels → internal servicing applications / external web site / electronic account trading platforms.

Transfer Agency Future State - Sonic as the centerpiece of our architecture





Getting started

We started relatively small, with a web customer self service project.

Focus on:

- Integration
- Information aggregation
- Encapsulating business rules in a set of reusable customer Services

Next: The 1st Phase of our Transfer Agencies Future State Architecture

- Implementation of our ECM / BPM tool in parallel with a supporting Services build out enabled by Sonic MQ/ESB
 - Initial set of Services focused around supporting Workflow through:
 - Legacy System Integration for Work Item attribute enrichment
 - Shareholder account Services and a respective canonical model
 - Foundational Services (i.e. Exception handling, retry, caching)

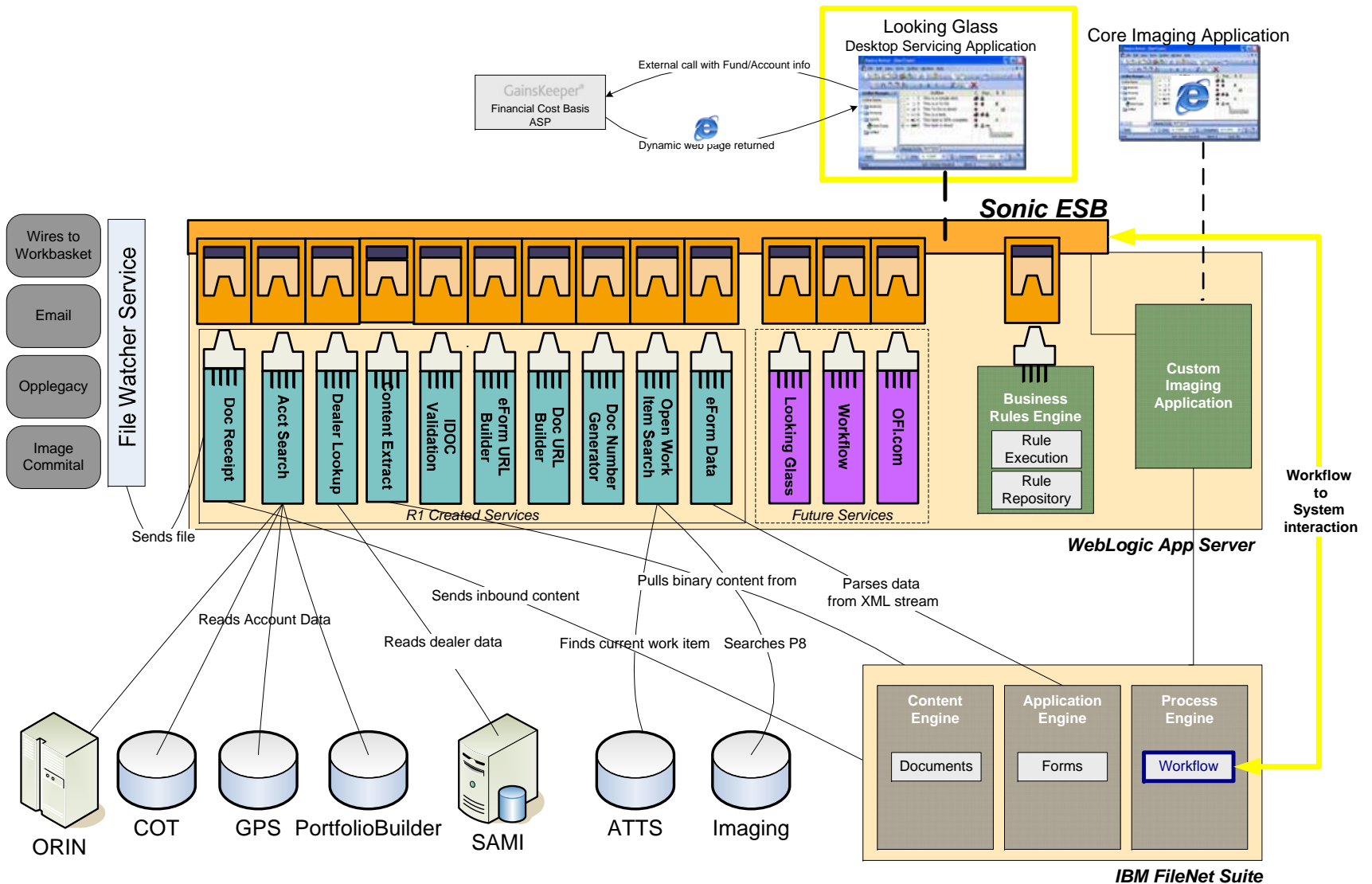


- We have introduced the Agile Development Methodology within certain groups at OppenheimerFunds.
- Amongst many other things, Agile preaches:
 - Incremental Production ready deliverables
 - Focus on business value
- SOA: A perfect match for Agile
 - Agile's incremental approach forces teams to decompose large functionality "stories" into more granularly focused deliverables.
 - Services fall out as a great way to deliver on this
 - Agile's intense focus on value highlights the reuse capabilities of SOA
 - We are getting a lot of traction around this with respect to building Shareholder maintenance Services that are exposed via internal and external (web) servicing applications

So how far along are we? ...



SOA / BPM – From the desktop all the way through to the Workflow engine





Some early pitfalls

- **Keep Deployment Environments Synchronized (Sizing can differ)**
 - Keep Development, QA and Production configurations in sync
 - Track Diffs: Port Numbers, Containers, FT & HA Clusters, Production passwords
 - Eliminates deployment headaches & trouble shooting during environment migration
 - Automate if possible (OFI built our own auto-deploy tool based on ant scripts)
- **Develop Canonical Model(s)**
 - The lack of an Information Model and Enterprise Schema owner causes service functionality duplication, re-use diminishes
 - Instead of consuming events, messages and data, developers gravitate back to RPC style services, old client-server models creep back in, tight coupling ensues
- **Employ Governance Early**
 - Standards for Auditing, Tracing, Logging, Request / Response enveloping
 - Value drops when duplicate Services emerge, Services lacking schemas, etc.
 - Differing Security Policies, Multiple binding models, Standards proliferation (SOAP vs.REST)
 - Early introduction of Registry/SOA Management Tools (but not too early – initially you can use spreadsheet and common sense)



Some key theme's to take away from this

- With BPM and SOA you can drive Operational Efficiency and Business Agility out of a legacy system environment.
 - Much easier to drive business value and build a business case (ROI) for this type of incremental approach
- A clean architecture can be developed when a BPM workflow solution is leveraged to drive the human side of processes vs. Sonic itineraries for system to system orchestration
- Matching Agile process with SOA is a win/win

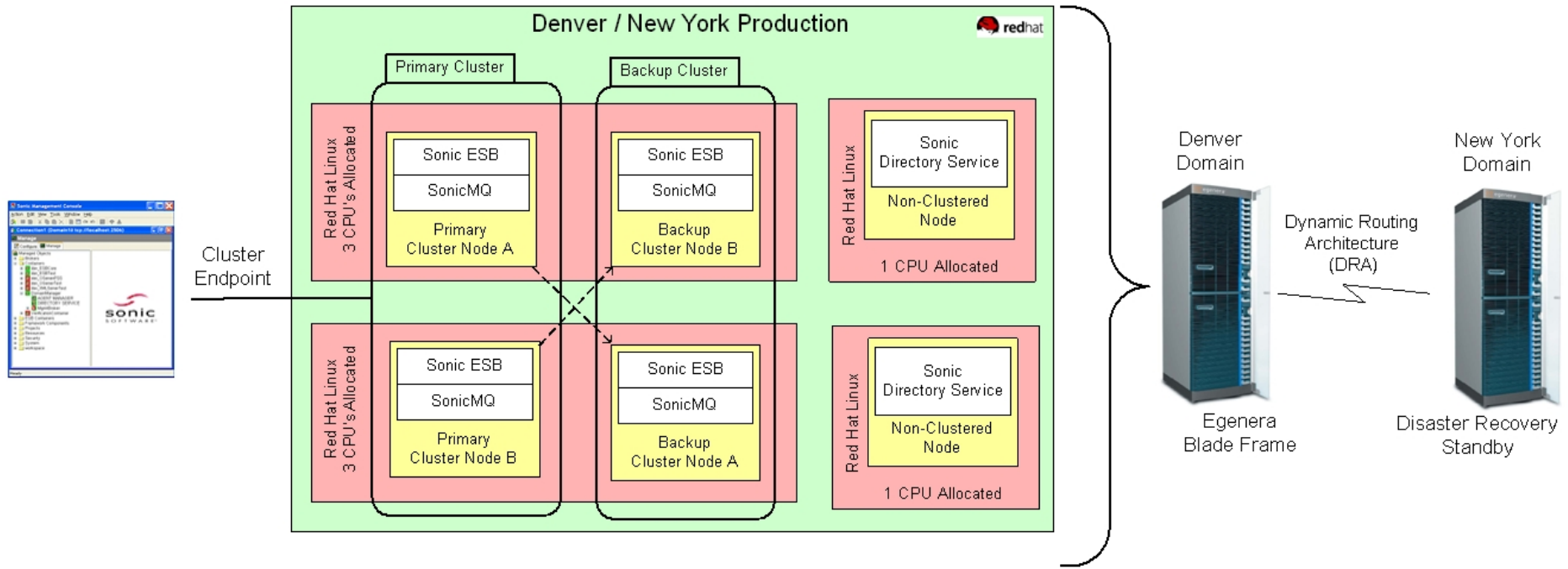


Questions



Deployment Architecture

Scaled down infrastructure exists for QA and Development



- Services deployed to New York Production domain are also deployed to Denver domain for DR purposes (and vice versa)
- Services in the DR site are only started in the event of the full failure of the primary Production environment