

GroupWise system and SMTP infrastructure design considerations

Lawrence Kearney

Enterprise and Workgroup Service Analyst

lawrence.kearney@earthlink.net

Session description

- Not a technical deep dive but an informal presentation on observed email service trends and the system architecture designs developed in response to them
- Third party security and traffic fencing technologies are numerous and will be referenced for conceptual purposes
- Most information is geared for large and enterprise size environments
- Designing and maintaining systems to compete with similar commercial service offerings

Servers, services, customers, etc ...

Servers being king is a dated concept

The local administrator's current environment consists of:

- Service providers
- Service consumers
- Servers as services

Do think in terms of what will make my “service” better

Do not only think in terms of what will make my “server” better

Do think about how to tune your server(s) to make your service better

Let's discuss some service issues

- Service performance
- Service efficiency
- Service stability
- Service feature sets
- Service cost of ownership
- Service planning and scaling
- Service standards

Influences leadership (service provider), staff, and customer
“**dissatisfaction**”

Finally some GroupWise service issues

Unmanaged desktop and device service consumers

- Fat POP/IMAP clients
- Thin POP/IMAP clients
- Device POP/IMAP clients

Services providers “consuming” GroupWise services

- BIS

Additional security concerns

- Cannot secure third party “local” message stores
- Cannot remotely manage or wipe unauthorized devices

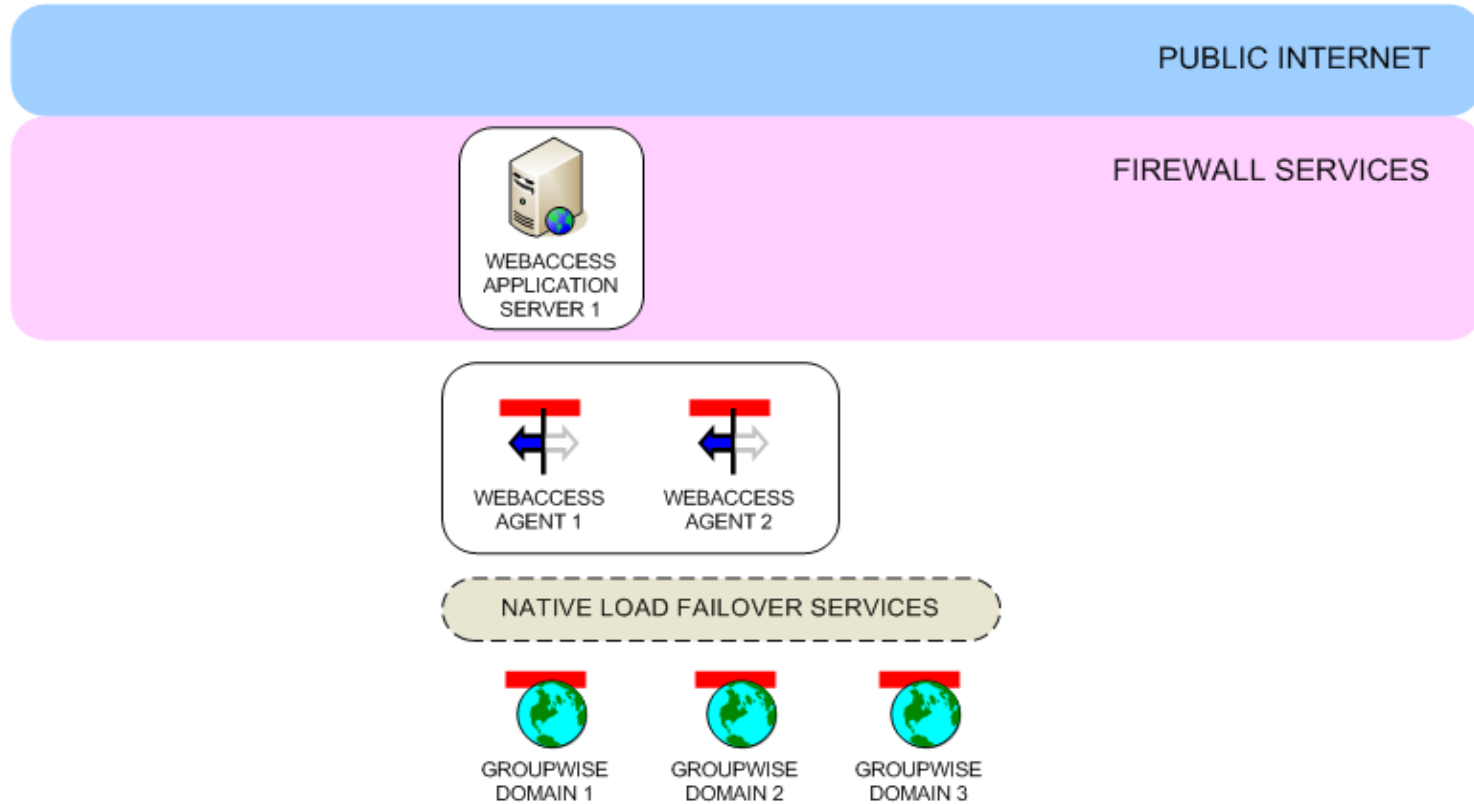
Eventually all will consume SMTP services and grossly consume POP/IMAP services

What can we do to improve services?

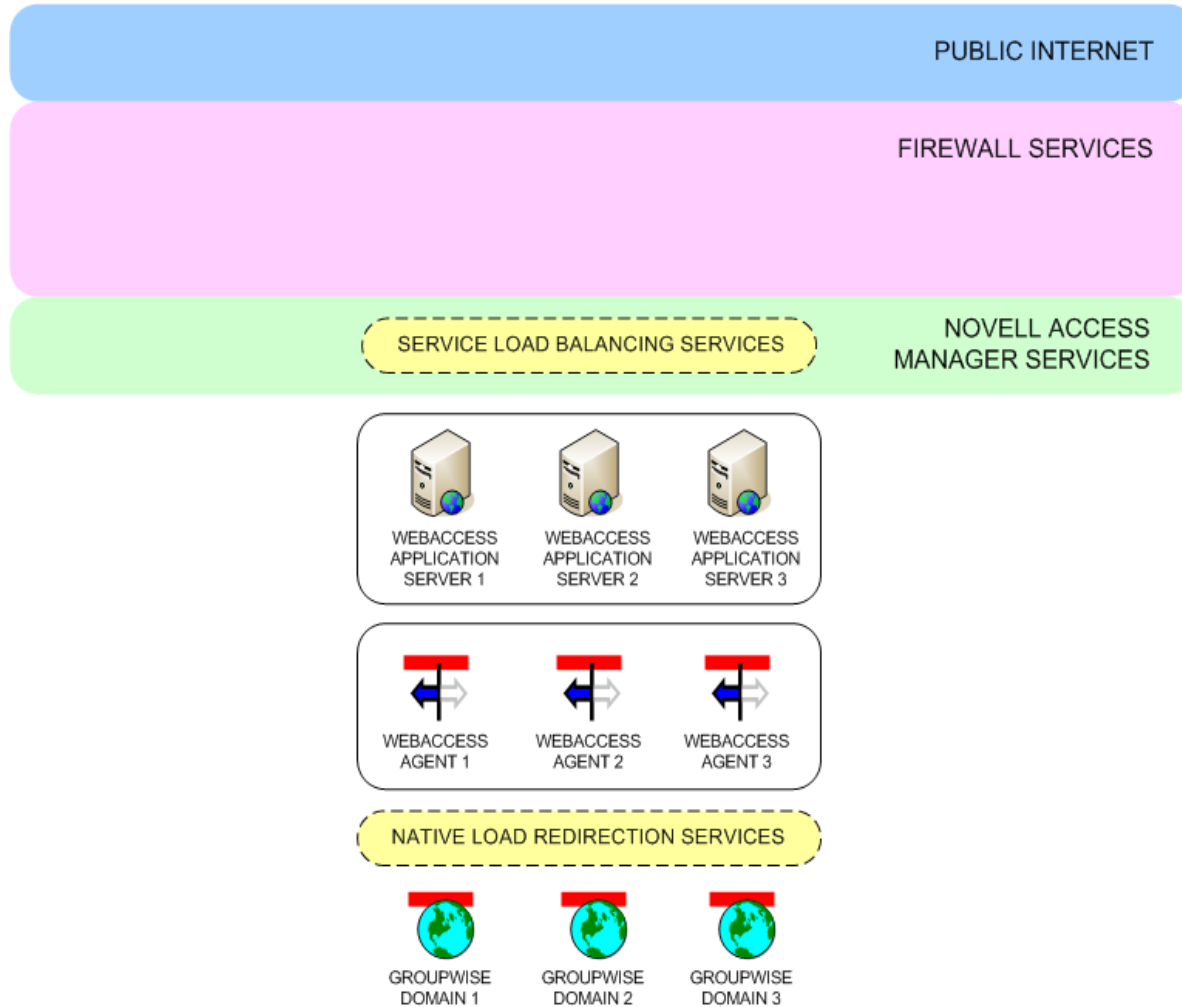
- Clever architecture and design choices
- Service redirection
- Service load balancing/management
- Service redundancy
- Service I/O fencing
- Service high availability
- Service demarcations (QoS really)
- Service virtualisation

Using a blended solution can both harden and improve the resiliency for your systems if designed well.

Legacy WebAccess example



Redesigned WebAccess example



Clever architecture and design choices

Discuss what service aspects and features your organisation is going to support prior to implementation

- Consider what your product can do natively and use it in your design
- Review service logs to determine service trends and metrics to shape implementation needs

Implement hardware between public and private services

- Helps relieve primary services of handling unnecessary I/O
- Does a better job than native GWIA unsolicited bulk email control

Load balancing using DNS

- Use “unbound” network addresses for services
- Load balance MX records
- When a single target has multiple names use native service load management
- When a single name has multiple targets use DNS load management

Continuing with the theme

Service redirection

- Not always available but can help “normalise” service traffic

Service load balancing

- Helps with economics of implementing solutions
- Software/dual purpose (lowers cost)
- Hardware (higher cost)

Service load management

- Can be “designed” into a system and managed with service consumption trend experience

Service redundancy

- Even in a small system always have more than one instance of mission critical services

Continuing with the theme

Service I/O fencing

- Native OS features
 - > CPU, logging, storage
- Service native features
- Infrastructure features
 - > Layer 2/3
 - > Design

Service high availability

- Helps with economics of implementing solutions
 - > Clustering solutions (often these include embedded HA options)
 - > Native service or OS elements

Continuing with the theme

Service demarcations

- Use to implement natural and intuitive user base boundaries
- Easy first step to implement
- Allows you to reap immediate benefits
- Limits the scope of planned/unplanned outages
- Leadership sees positive aspects
- Document and publicise service access “protocols” and network addresses

Service virtualisation

- Helps with economics of implementing solutions
- Use it where appropriate

So what do we get if we consider this stuff?

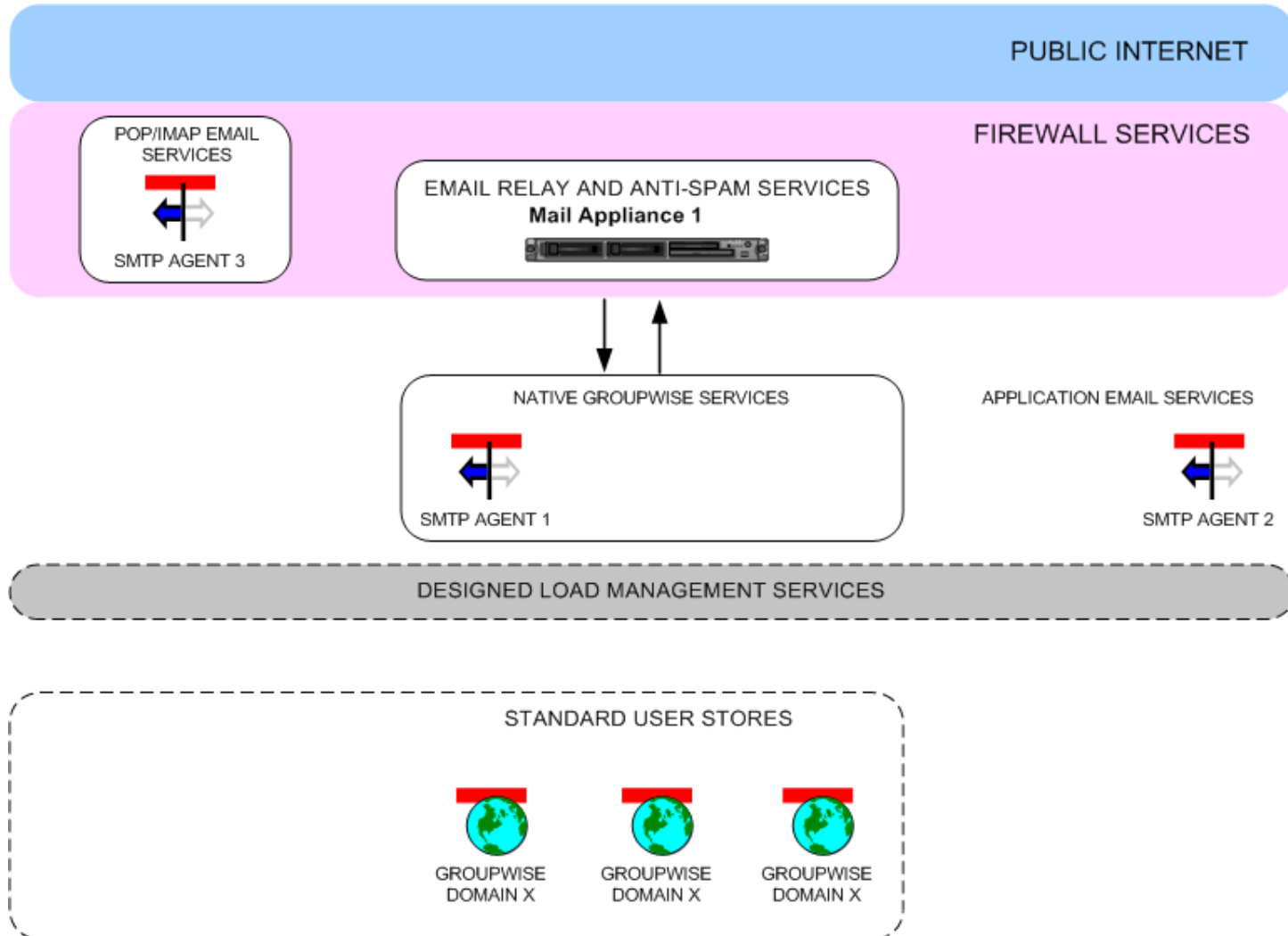
- Service performance improvements
- Service availability improvements
- Service capacity monitoring improvements
- Service capacity management improvements
- Service feature set expansions

Influences leadership, staff, and customer “**satisfaction**”

MCG environment nuances (maybe yours too!)

- Native GroupWise services
- Application Email services
- Non-authenticated email services (internal openish relay)
- Authenticated email services
- Unmanaged email client services
 - OMG if you just watched those logs

Legacy SMTP infrastructure example



MCG GroupWise service architecture decisions

GroupWise SMTP services

- GWAHA services are used everywhere

GroupWise Internet Agents

- Native GroupWise ACLs are applied where applicable
- Agents are chained for fail over where applicable
- Multiple mail relay targets are applied
- Never allow “true” open relay configurations
- Utilise multiple domain names
- Are native and non-native load balanced wherever possible

Gateway domains

- No regular users live here
- Service specific gateways are located here

Continuing with the theme ...

User post offices

- No unmanaged POP/IMAP connectivity to POAs

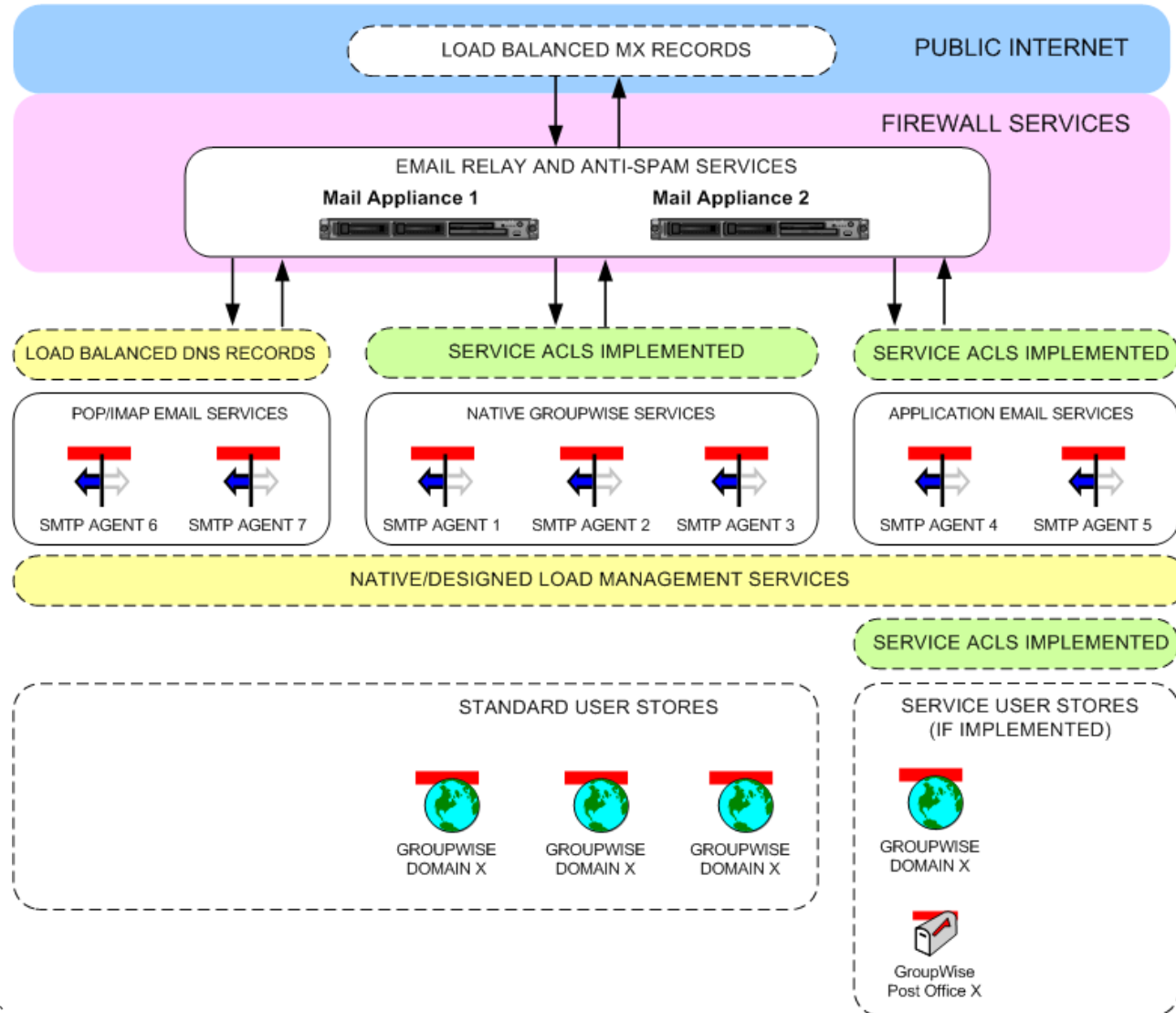
Service domains

- Application specific SMTP gateways
- IDM services connect here

Service post offices

- No regular users live here
- Managed third party client ingress points
- Application accounts
- SMTP authentication accounts

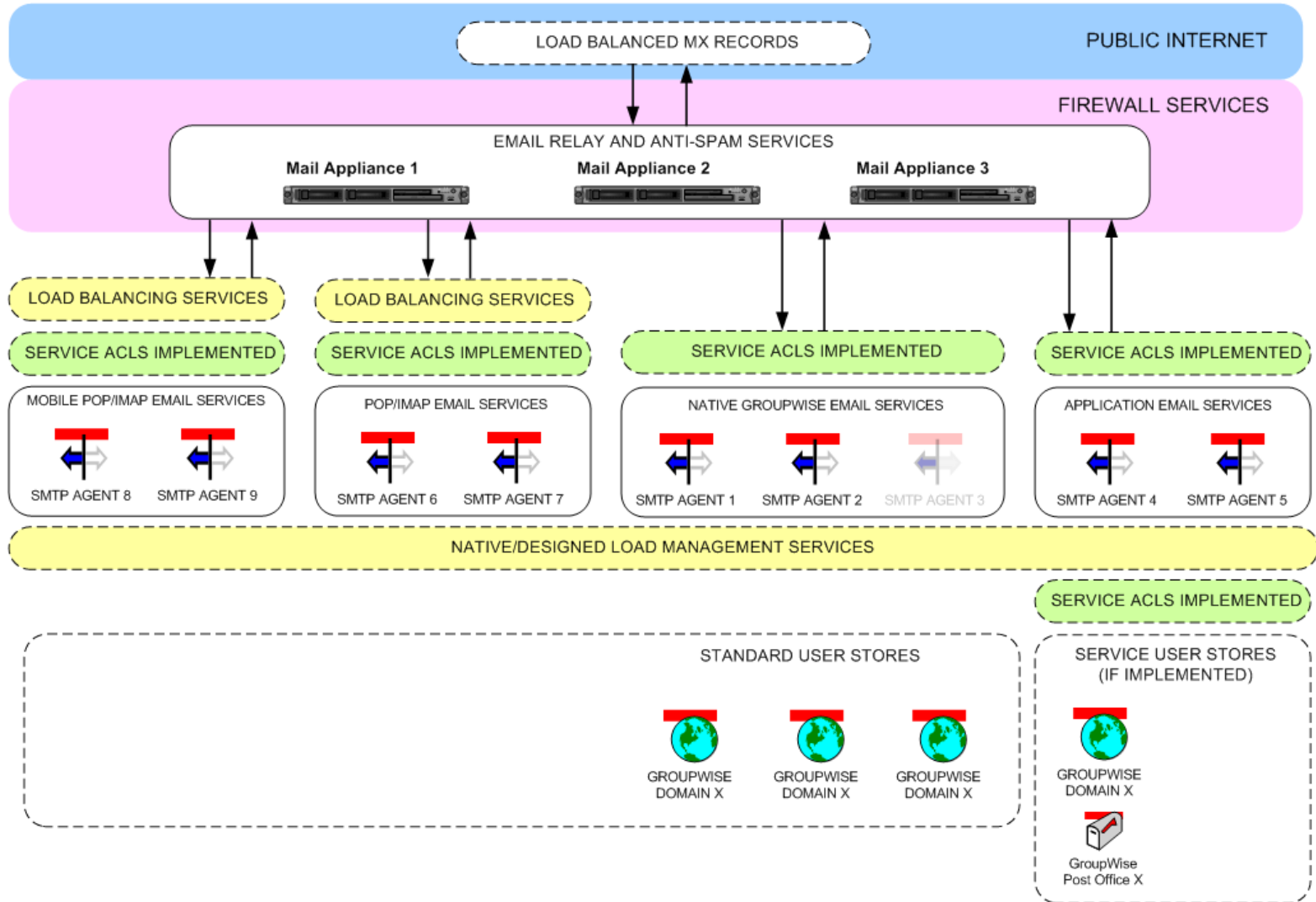
Redesigned SMTP infrastructure example



Continuing with the theme ...

Additional infrastructure design changes are planned

Final SMTP architecture example



Unpublished Work of Novell, Inc. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary, and trade secret information of Novell, Inc. Access to this work is restricted to Novell employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Novell, Inc. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. Novell, Inc. makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for Novell products remains at the sole discretion of Novell. Further, Novell, Inc. reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.



Novell.[®]