Microsoft

tech·days

Hong Kong | 2013

You make
the difference

Microsoft

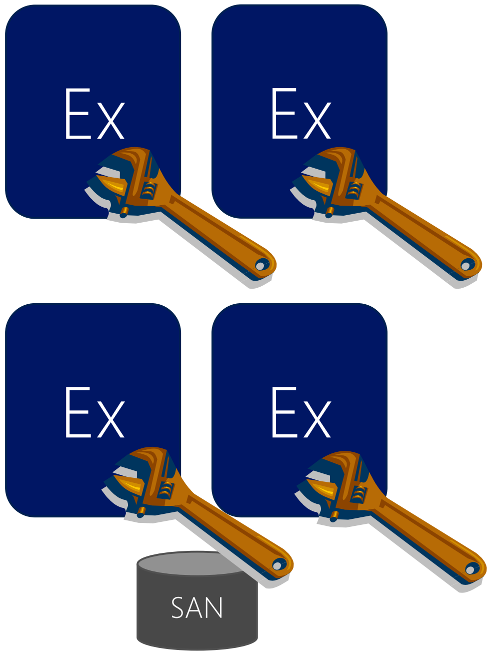
Exchange Server 2013 Architecture Overview

Scott Schnoll
Principal Technical Writer
Microsoft Corporation

Agenda

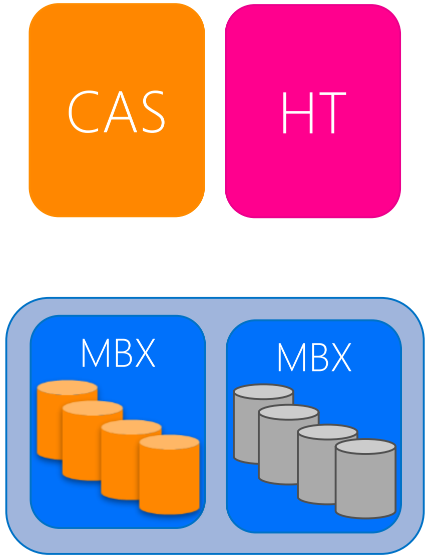
- Exchange Server Evolution
- Architecture Changes
- Client Access Server Role
- Mailbox Server Role

Exchange: Past, Present and Future



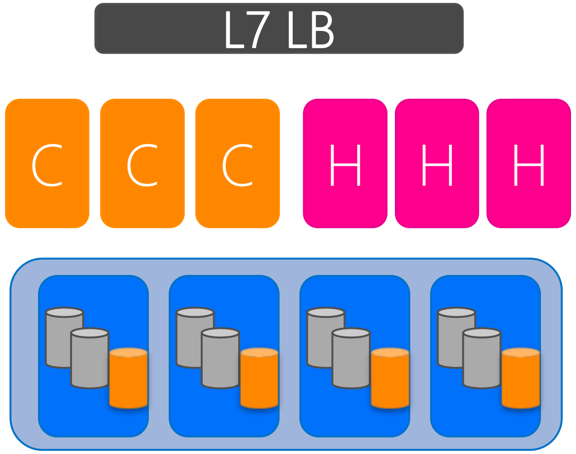
- Role differentiation through manual configuration
- Backups and hardware solutions for "reliability"

2000/2003



- Separate roles for deployment & segmentation
- Support cheaper storage

2007



- Separate HA solution per role
- Introduction of the DAG
- Support for Hybrid deployments

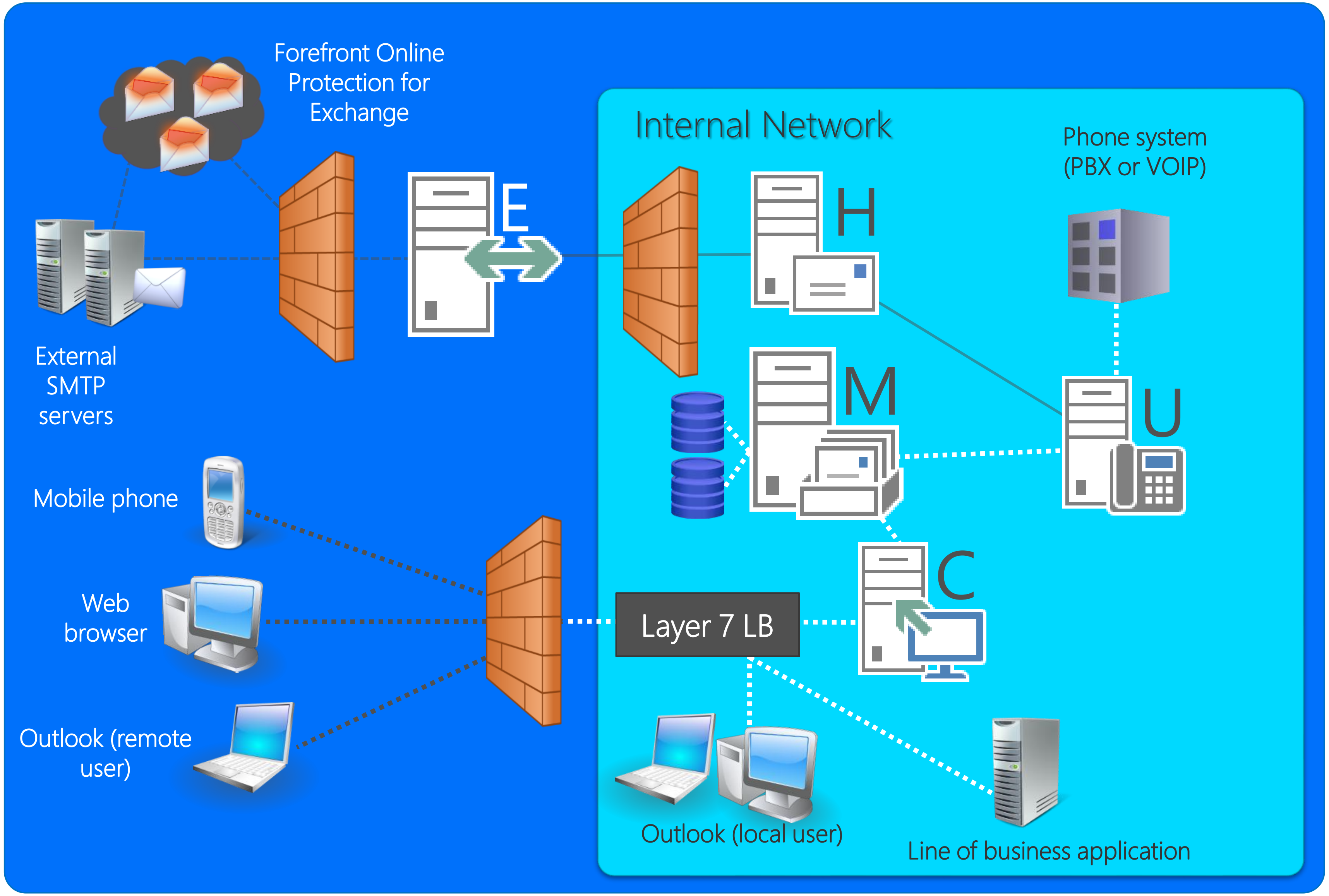
2010

?

Previous Server Role Architecture

- 5 server roles

- Tightly-coupled in terms of
  - versioning
  - functionality
  - user partitioning
  - geo-affinity

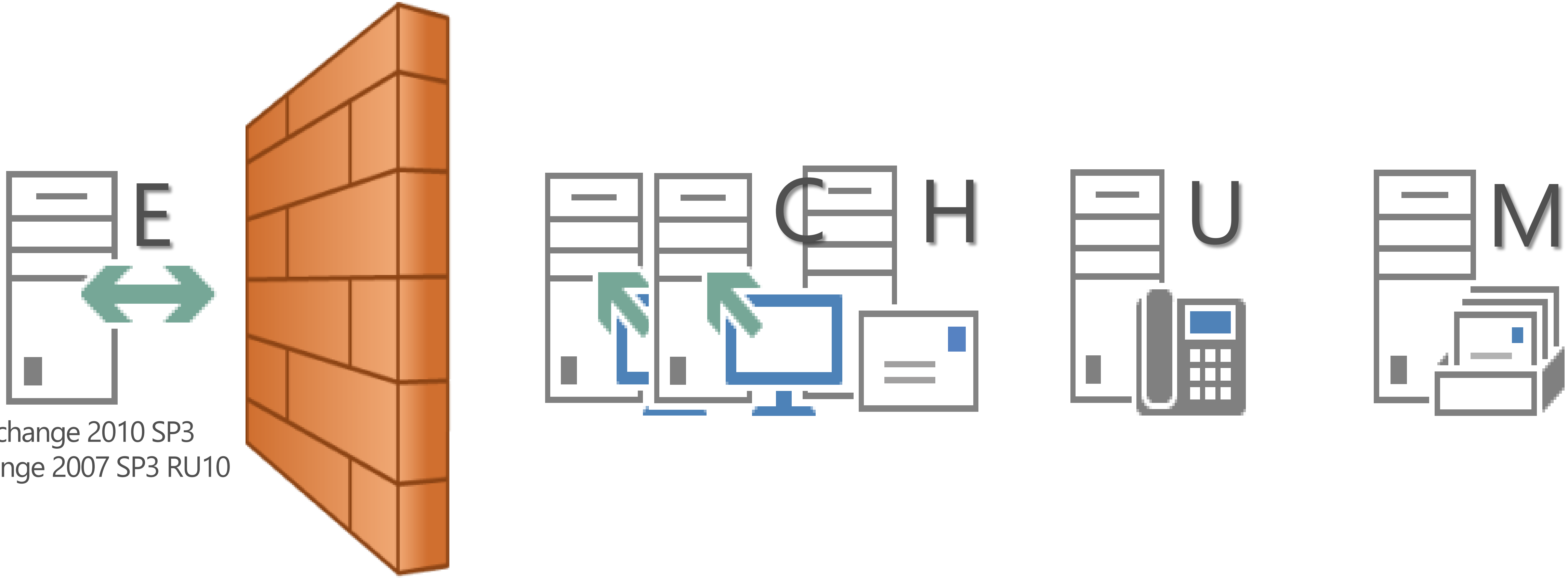


Challenges with Legacy Model

- Exchange deployments can be complicated
- Load balancing can be difficult and expensive
- Hardware can go unutilized or under-utilized
- Too many namespaces required

Evolution of Server Roles

Evolution of Server Roles



Exchange 2010 SP3
Exchange 2007 SP3 RU10

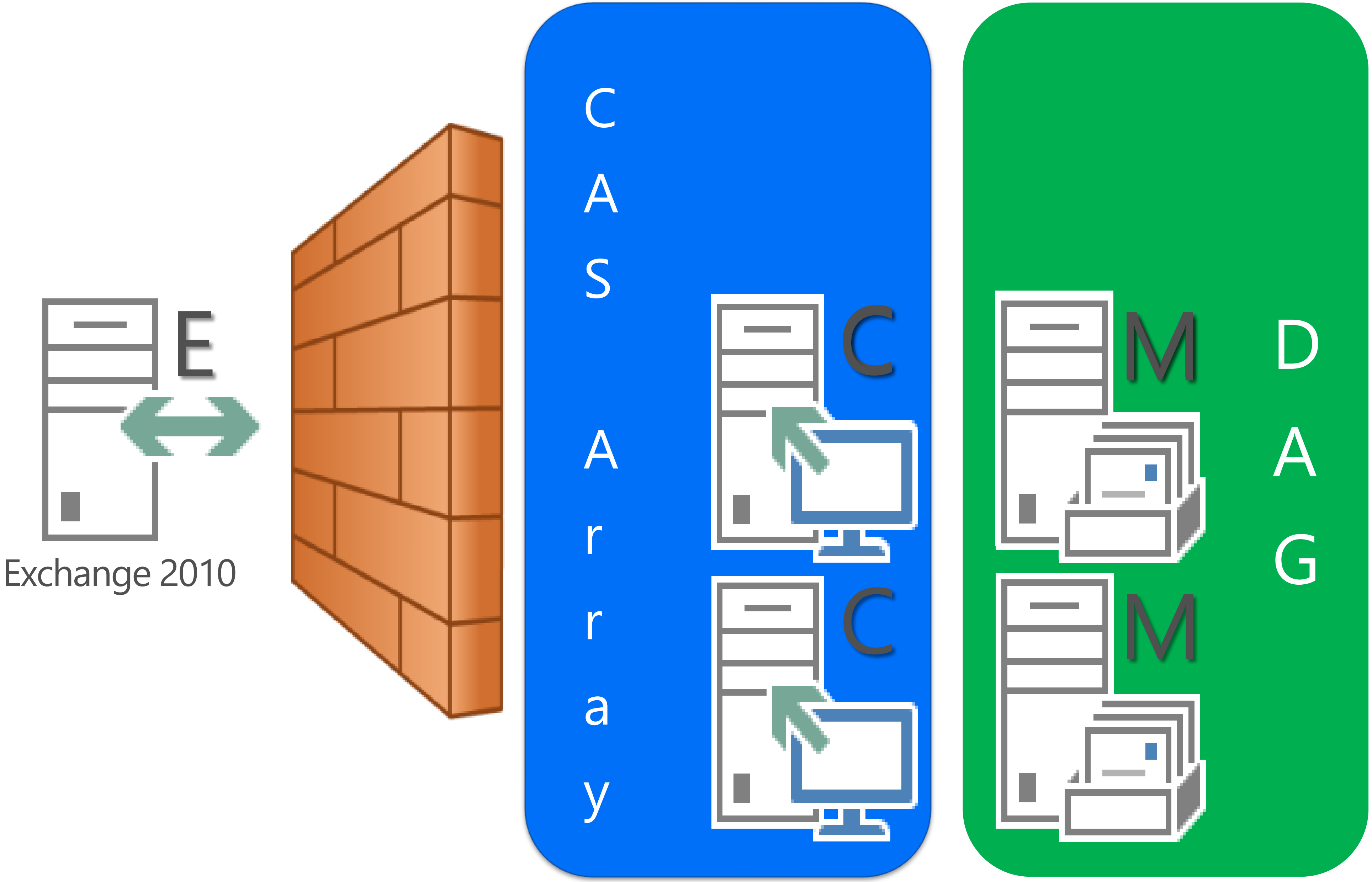
Client Access Server Role

- Thin, stateless (protocol session) server that includes:
  - Client access protocols (HTTP, POP, IMAP)
  - SMTP proxy
  - UM call router
- Exchange-aware proxy server
  - Understands requests from client protocols
  - Supports proxy and redirection logic for client protocols

Mailbox Server Role

- Server that processes, renders and stores Exchange data
- Includes components previously found in separate roles (CAS, Hub, UM)
- Connectivity to user's mailbox is always provided by the protocol stack on the Mailbox server hosting the active database copy

Evolution of Server Roles

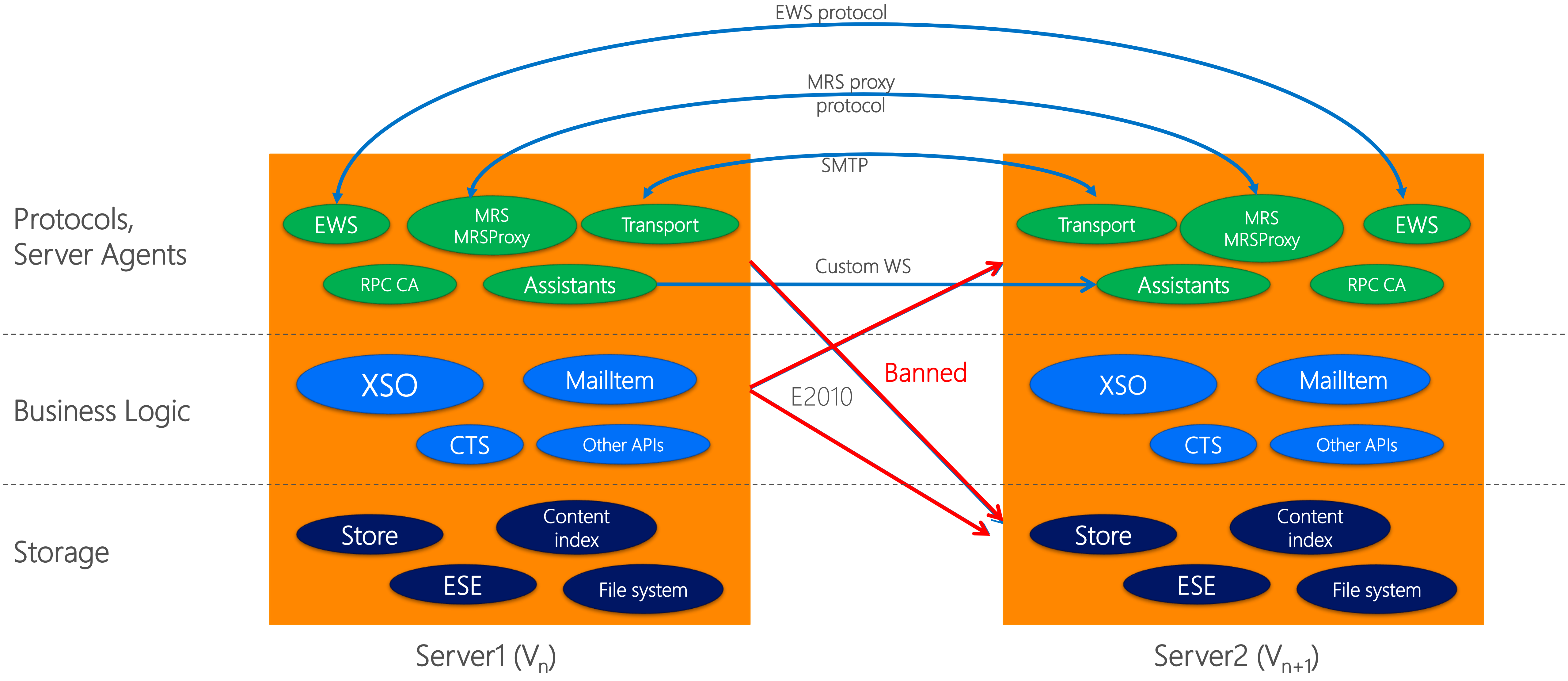


Architectural Changes

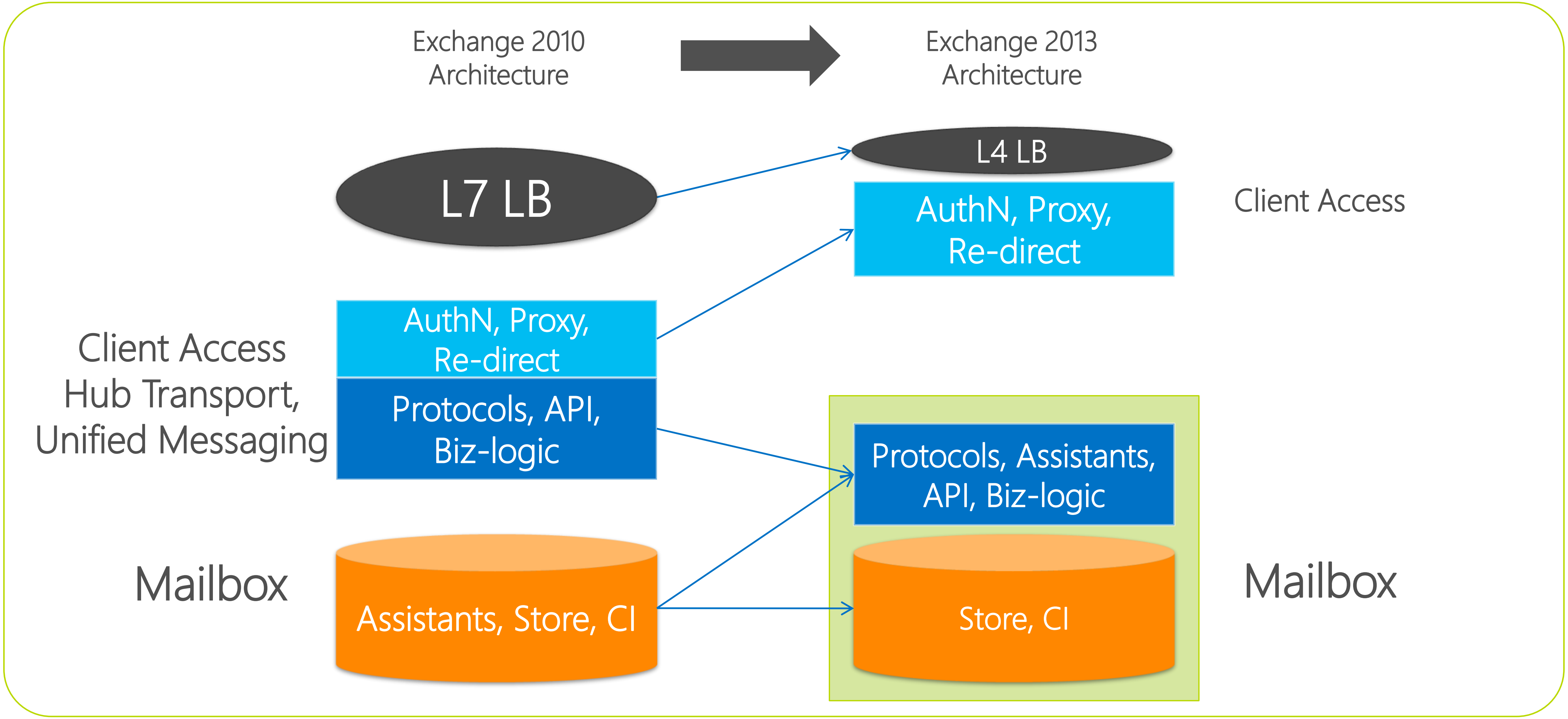
Architecture Theme and Benefits

- Use Building Blocks to facilitate deployments at all scales
  - Server role evolution
  - Network layer improvements
  - Versioning and inter-op principles
- Numerous Benefits
  - Hardware efficiency
  - Deployment simplicity
  - Cross-version inter-op
  - Failure isolation

Key Tenet: *Every Server is an Island*



Functional Differences



Exchange 2010 Architecture

Exchange 2013 Architecture

L7 LB

L4 LB

AuthN, Proxy, Re-direct

Client Access

Client Access Hub Transport, Unified Messaging

AuthN, Proxy, Re-direct

Protocols, API, Biz-logic

Protocols, Assistants, API, Biz-logic

Mailbox

Assistants, Store, CI

Store, CI

Mailbox

Client Access Server Role

Client Access Server Role

- Domain-joined machine in the internal Active Directory forest
  - Thin, stateless (protocol session) server
- Comprised of three components:
  - Client access protocols (HTTP, IMAP, POP)
  - SMTP
  - UM Call Router
- Exchange-aware proxy server
  - Understands requests from different protocols (OWA, EWS, etc.)
  - Contains logic to route specific protocol requests to their destination end-point
  - Supports proxy and redirection logic for client protocols

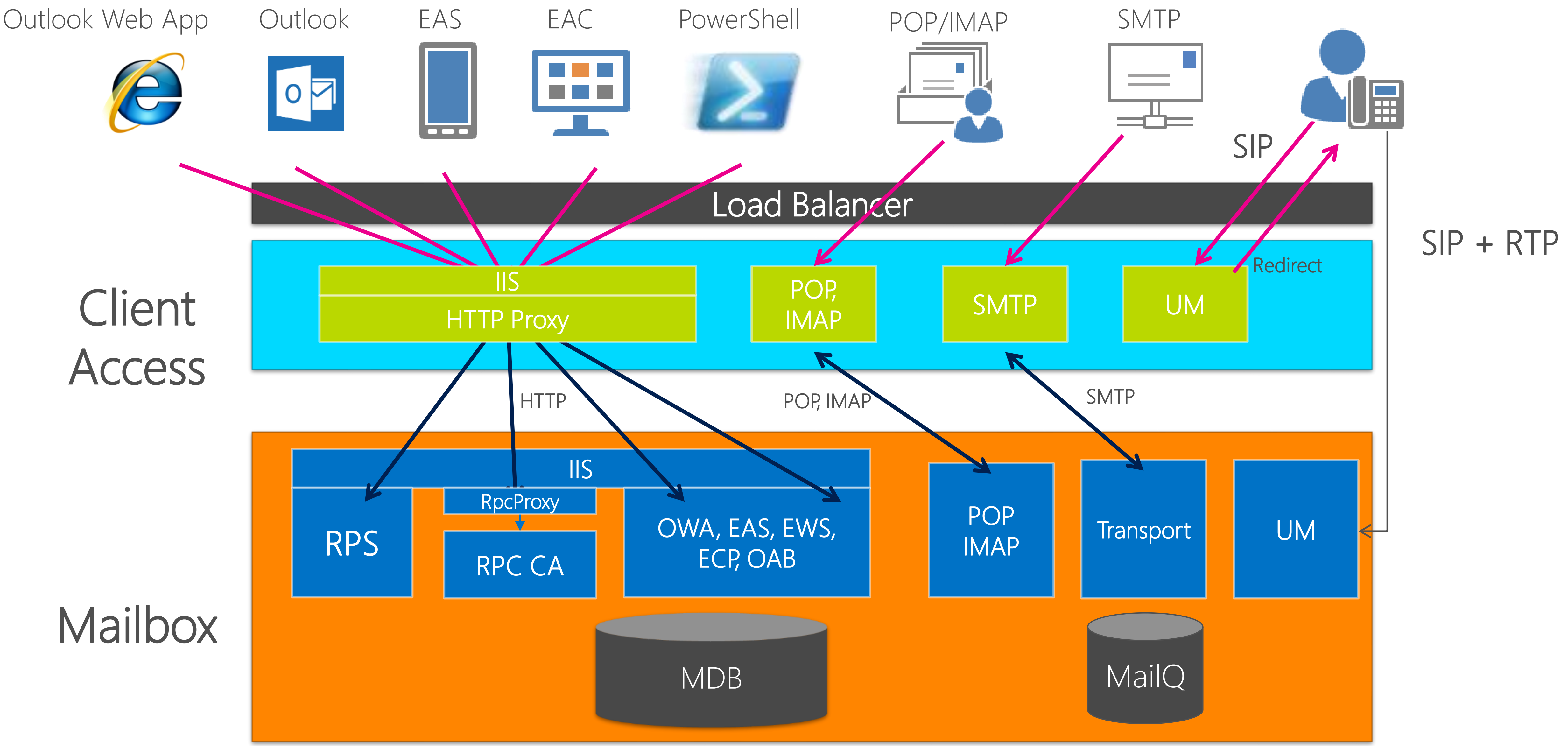
Client Access Array

- A group of CAS organized in a load-balanced configuration
- Designed to work with TCP affinity (aka, layer 4 LB)
  - Does not require application-level session affinity (aka, layer 7 LB)
- Provides a unified namespace and authentication
  - Similar to Exchange 2010 in terms of providing a unified endpoint for client connectivity and authentication

Outlook Connectivity Changes

- Exchange 2013 supports RPC/HTTP only
  - RPC/TCP not supported
- Benefits
  - Simplifies the protocol stack
  - Provides a reliable and stable connectivity model
  - RPC session is always on Mailbox server hosting active copy
  - Eliminates need for RPC CAS Array and RPC CAS Array namespace(s)
  - Eliminates interruptions like "*The Exchange administrator has made a change that requires you to quit and restart Outlook*" during mailbox moves or *overs

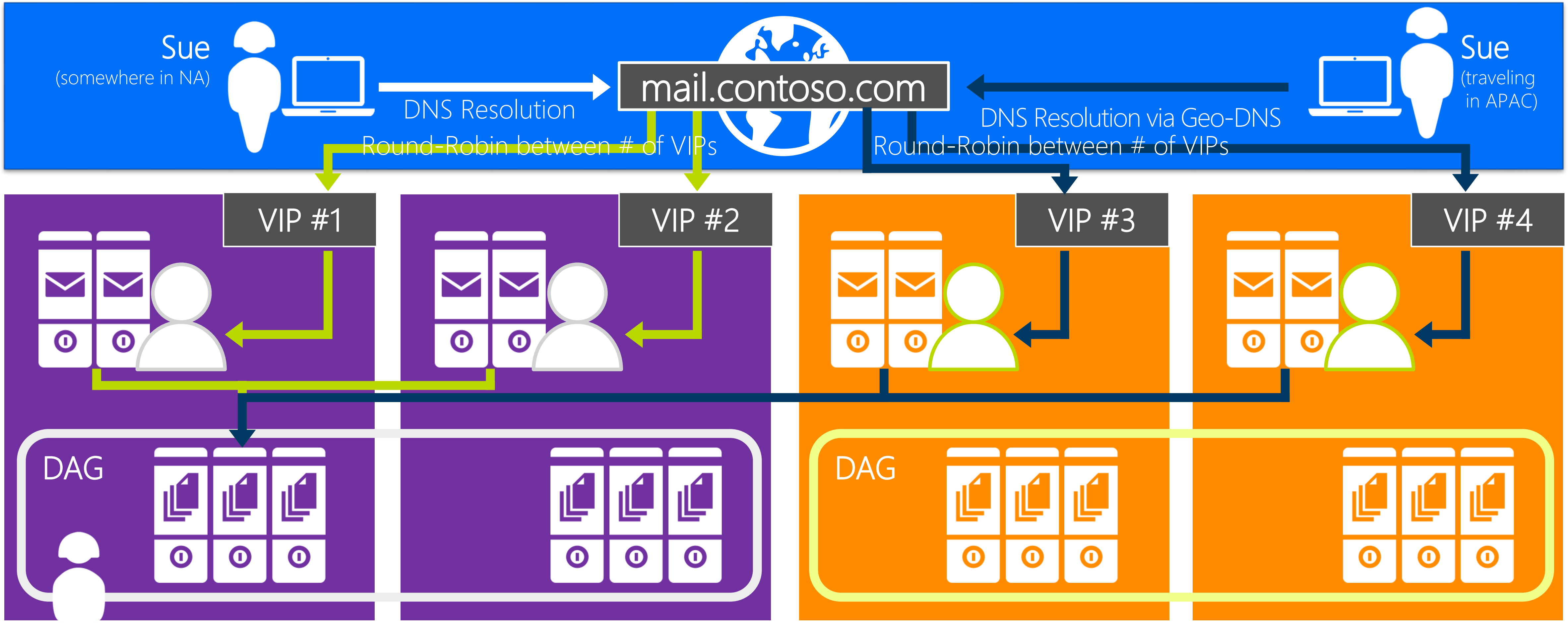
Client Protocol Flow in Exchange 2013



Namespace Simplification

- Exchange 2013 supports new namespace configurations
  - Multiple namespaces supported
  - Single global namespace supported
    - Can be used in coexistence with Exchange 2010

Single Common Namespace



Front End Transport Service

Front End Transport Service

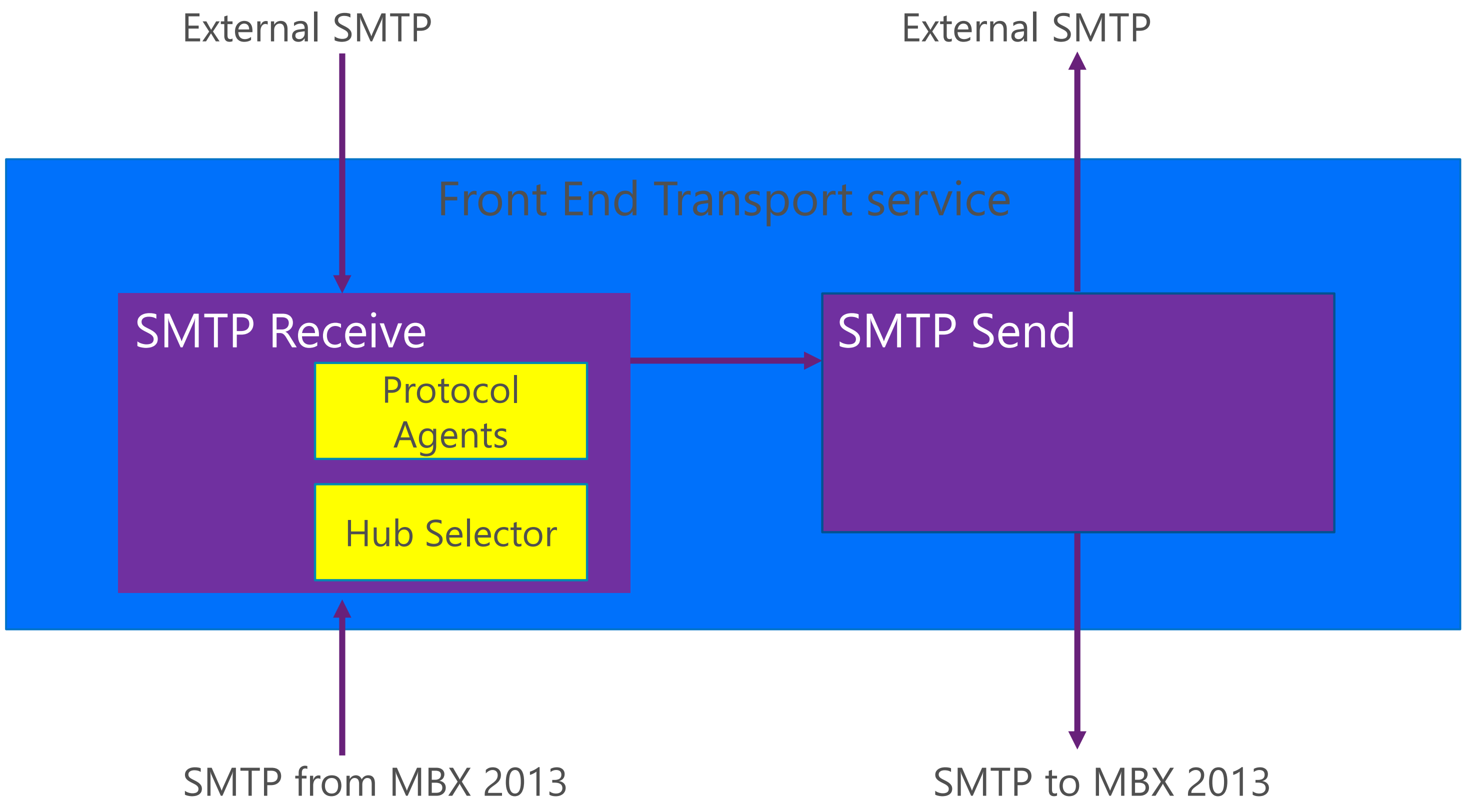
- Handles all inbound and outbound external SMTP traffic for the organization
  - Listens on TCP25 and TCP587 (two receive connectors)
    - Also acts as endpoint for client traffic
  - Does not replace the Edge Transport Server role
  - Functions as a layer 7 proxy and has full access to protocol conversation
  - Does not queue mail locally, and is stateless
  - If enabled, all outbound traffic appears to come from CAS 2013

Front End Transport Service

- Network protection – centralized, load-balanced egress/ingress point for the organization

- Mailbox locator – avoids unnecessary hops by determining the best MBX 2013 to deliver the message

Front End Transport Service



Mail Flow

Inbound Mail Flow

1. FET accepts initial SMTP connection
2. After DATA command is issued, FET determines the next destination for the recipients in the message
3. FET starts the SMTP proxy session to the appropriate destination

Outbound Mail Flow

1. MBX 2013 determines if mail recipient is a remote destination and selects a FET within local site when the FrontEndProxyEnabled parameter on Send Connector is set to $true
2. MBX 2013 connects to FET and initiates SMTP conversation
3. FET proxies outbound connection to appropriate destination

Mail Routing

- FET uses entry point routing with delivery groups
  - DAG, Mailbox Server, AD Site
- Bifurcation does not occur on FET
  - Only one DAG or Mailbox server is selected, regardless of number of recipients
- Server selection within delivery group based on recipient type
  - If message only has a single mailbox recipient, select MBX server within delivery group based on proximity of AD site
  - If multiple mailbox recipients, select MBX server in closest delivery group, factoring in site proximity
  - If there are no mailbox recipients (DG, MEUs, etc.), select a random MBX 2013, giving preference to local AD site

Mailbox Server Role

Mailbox Server Role

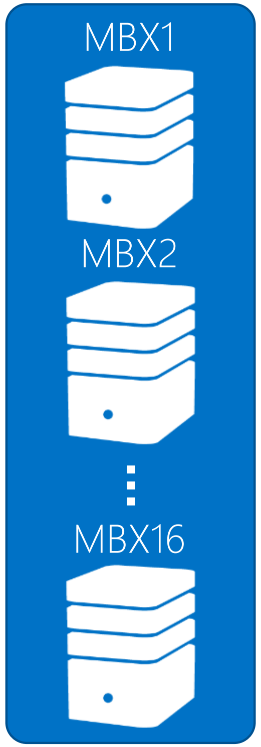
- Server that hosts components that process, render and store Exchange data

- Includes components previously found in separate roles

- Connectivity to a mailbox is always provided by the protocol instance on the server hosting the active database copy

Database Availability Group

- Collection of servers that form a unit of high availability
- Boundary for replication and *over
- DAG members can be in different sites
- Can have a maximum of 16 Mailbox servers



Mailbox Server Role Changes

- Managed Store
- IOPS Reductions
- Transport Changes
- Modern Public Folders

Managed Store

Exchange Information Store

- Previously a single monolithic process
- Unmanaged code
- Very nested code that made it difficult to debug

Managed Store

- Store service/process (Microsoft.Exchange.Store.Service.exe)
  - Microsoft Information Store service
  - Manages worker process lifetime based on mount/dismount
  - Logs failure item when store worker process problems detected
  - Terminates store worker process in response to “dirty” dismount during failover
- Store worker process (Microsoft.Exchange.Store.Worker.exe)
  - One process per database, RPC endpoint instance is database GUID
  - Responsible for block-mode replication for passive databases
  - Fast transition to active when mounted
  - Transition from passive to active increases ESE cache size 5X

Microsoft Exchange Replication service

- Replication service process (MSExchangeRepl.exe)
  - Detecting unexpected database failures
  - Issues mount/dismount operations to Store
  - Provides administrative interface for management tasks
  - Initiates failovers on failures reported by ESE, Store, and Responders

New ESE Cache Management Algorithm

- Allocates 25% of RAM for worker process ESE cache
  - This is referred to as the max cache target
  - Amount allocated to each store worker process based on number of hosted database copies and value of MaximumActiveDatabases
  - Static amount of cache allocated to passive and active copies
- Store worker process will only use max cache target when copy is active
  - Passive database copies allocate 20% of max cache target
- Max cache target computed at service startup
  - Restart Store service process when adding/removing copies or changing value of MaximumActiveDatabases

IOPS Reductions

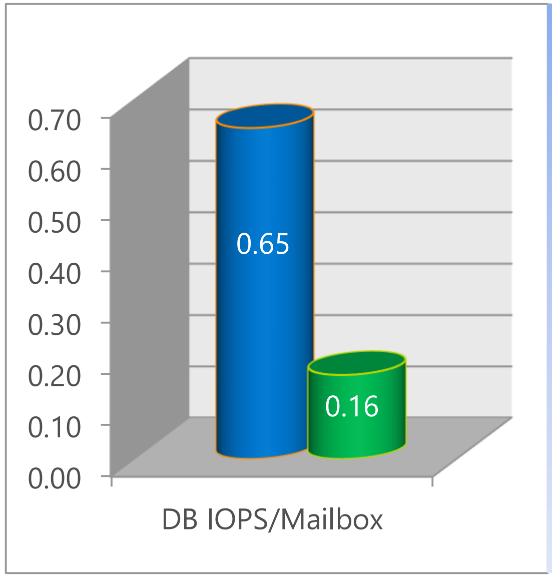
IOPS Reductions

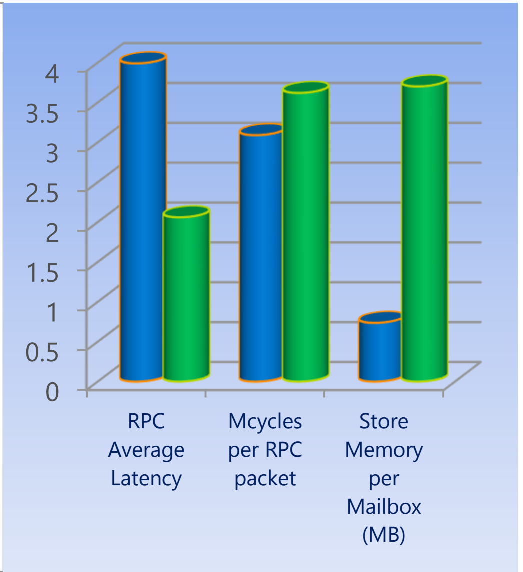
- Improvements to logical contiguity of store schema
  - Property blobs are used to store actual message properties
  - Several messages per page means fewer large IOs to retrieve message properties
  - Use of long-value storage is reduced, though when accessed, large sequential IOs are used
- Reduction in passive copy IO
  - 100MB checkpoint depth reduces write IO
  - Transaction log code refactored for faster failover

IOPS Reductions

Element	Exchange 2007	Exchange 2010/2013
Physical Contiguity (ESE)	Poor physical contiguity of leaf pages. Hence many, small-sized IOs (1 for each page)	Excellent physical contiguity of leaf pages. So fewer, large-sized IOs, spanning N pages
Logical Contiguity (Store)	Headers for each folder kept in separate table. So many, small-sized IOs spread over many tables	Single message table for an entire mailbox. (Property blobs used to store actual message properties, large blob in LV) Several messages/page, fewer large IOs to retrieve message properties in view)
Temporal Contiguity (View)	All views and indexes updated each time a mail is delivered. So many, small-sized IOs spread over time	Views and indexes updated only when they are accessed by user. So fewer, large-sized IOs are done together

E14 vs. E15: DITL Performance Comparison





- 48 | 76% reduction in disk IOPS
- 18 | 41% reduction in Average RPC Latency
- 17 | 34% increase in CPU per RPC processed
- ~4x increase in Store memory overhead

LoadGen Simulation – 10 DBs/1000 users

Two profiles: Online and Cached (Default/Optimized)

Perf gains are not free – increase in CPU and memory

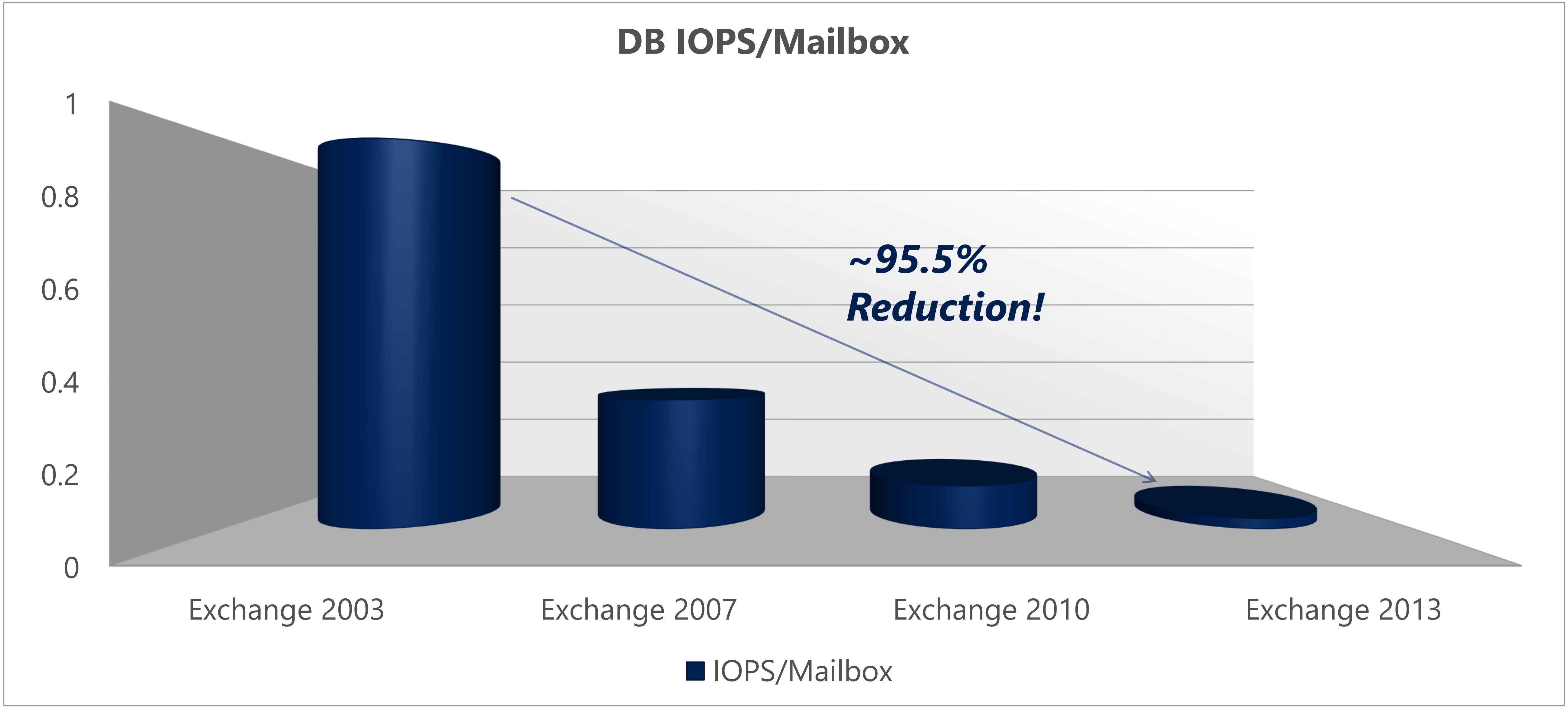
CPU increase is factor of optimizing for two-socket servers and moving to multi-process architecture

> Enables us to scale out using multi-core processors without having to cross processor bridge to access shared L2 cache
>
> Some CPU overhead comes from using managed code

Memory increase is also factor of multi-processor architecture

> Most of the memory is in small and large object heaps in .NET primarily used for object allocation and cleanup

IOPS Reductions



Transport Changes

Transport Changes

- Transport on Mailbox server is comprised of three services:
    - Microsoft Exchange Transport - Stateful and handles SMTP mail flow for the organization and performs content inspection
    - Microsoft Exchange Mailbox Transport Delivery - Receives mail from the Transport service and deliveries to the mailbox database
    - Microsoft Exchange Mailbox Transport Submission - Takes mail from the mailbox databases and submits to the Transport service

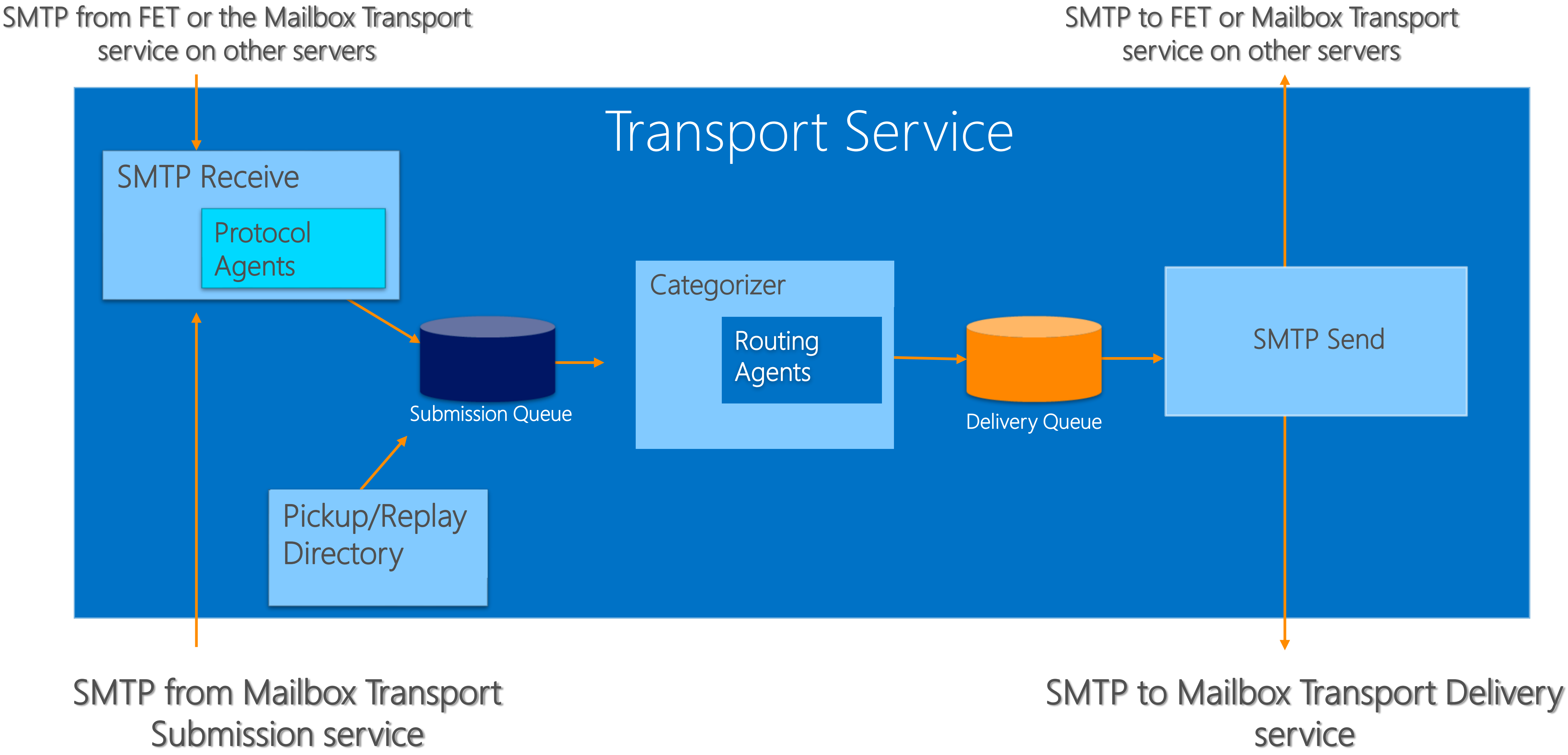
Transport Changes

- Transport has the following responsibilities
  - Receives all inbound mail to the organization
  - Submits all outbound mail from the organization
  - Handles all internal message processing such as transport rules, content filtering, and antivirus
  - Performs mail flow routing
  - Queue messages
  - Supports SMTP extensibility

Mailbox Transport

- Two separate services for mail submissions (from the store) and mail delivery (from the Transport service)
  - Mailbox Assistant and Store Driver combined
- Leverages SMTP (encrypted) for communication with the Transport component and TCP465 for inbound traffic
- Leverages local RPC for delivery to store
- Is stateless and does not have a persistent storage mechanism

Transport Service Architecture



SMTP from FET or the Mailbox Transport service on other servers

SMTP to FET or Mailbox Transport service on other servers

Transport Service

SMTP Receive

Protocol Agents

Submission Queue

Categorizer

Routing Agents

Delivery Queue

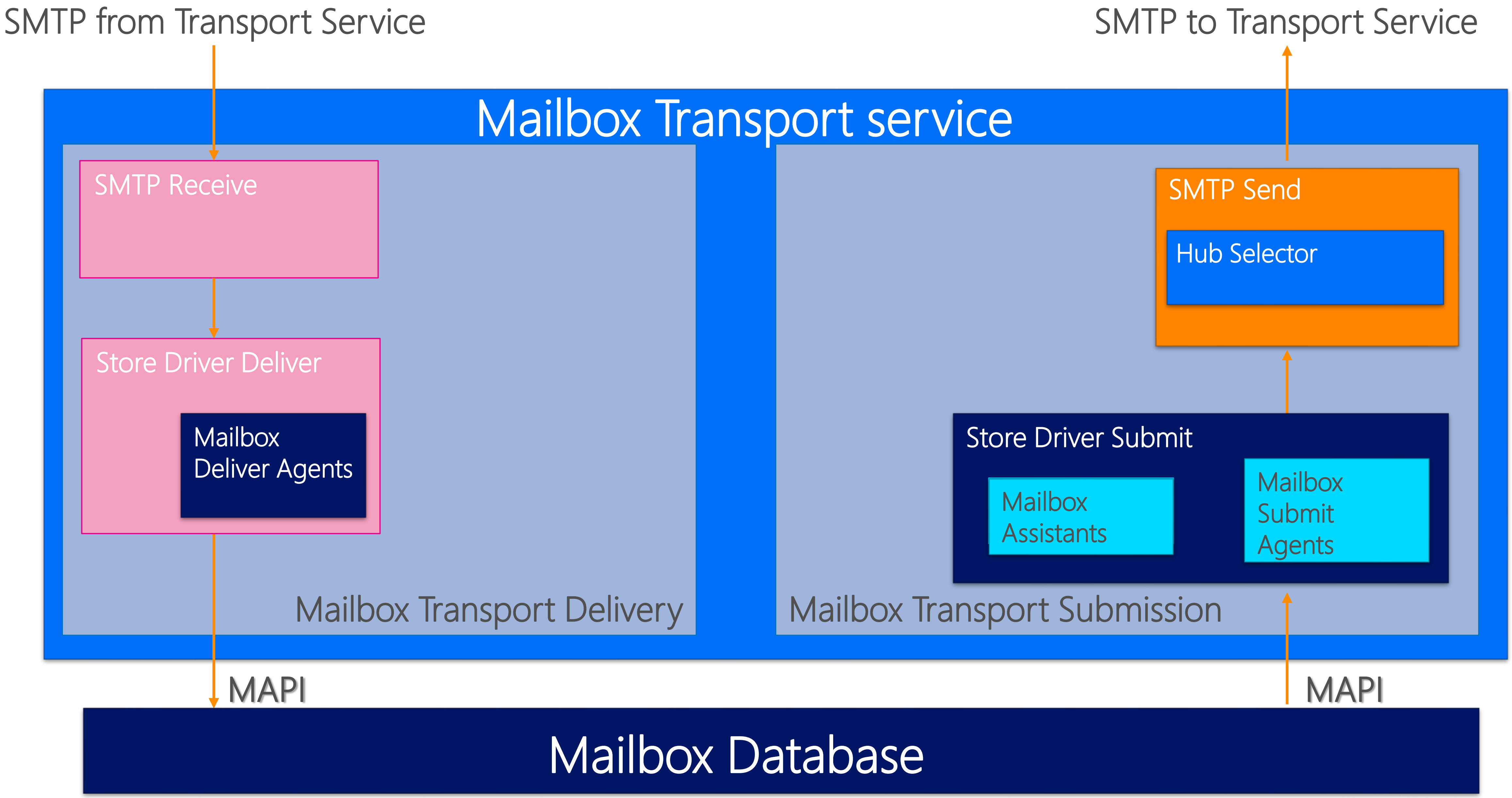
SMTP Send

Pickup/Replay Directory

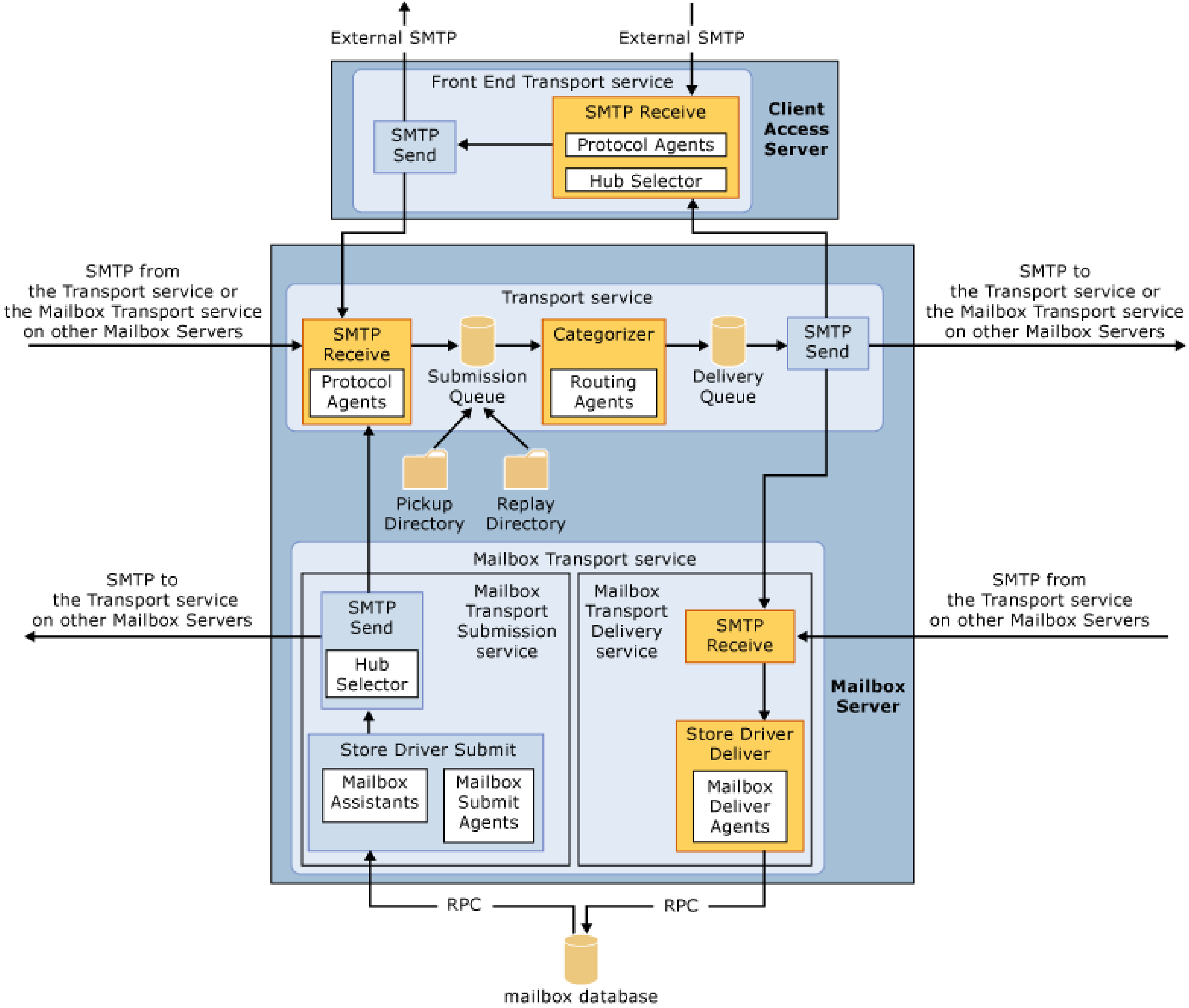
SMTP from Mailbox Transport Submission service

SMTP to Mailbox Transport Delivery service

Mailbox Transport Service Architecture



Exchange 2013 Transport Architecture


External SMTP
External SMTP
Front End Transport service
SMTP Send
SMTP Receive
Protocol Agents
Hub Selector
Client Access Server
SMTP from the Transport service or the Mailbox Transport service on other Mailbox Servers
Transport service
SMTP Receive
Protocol Agents
Submission Queue
Categorizer
Routing Agents
Delivery Queue
SMTP Send
SMTP to the Transport service or the Mailbox Transport service on other Mailbox Servers
Pickup Directory
Replay Directory
Mailbox Transport service
SMTP to the Transport service on other Mailbox Servers
SMTP Send
Hub Selector
Mailbox Transport Submission service
Mailbox Transport Delivery service
SMTP Receive
SMTP from the Transport service on other Mailbox Servers
Mailbox Server
Store Driver Submit
Mailbox Assistants
Mailbox Submit Agents
Store Driver Deliver
Mailbox Deliver Agents
RPC
RPC
mailbox database

Modern Public Folders

Modern Public Folders

- Public folders based on the mailbox architecture
  - Single-master model
  - Hierarchy is stored in a PF mailbox (one writeable)
  - Content stored in one or more mailboxes
  - The hierarchy folder points to the target content mailbox
- Because it's a mailbox, it's in a mailbox database…thus,
  - High availability achieved through continuous replication
  - No separate replication mechanism

Modern Public Folders

- Users connect to home Public Folder mailbox first
  - Should be located near their primary mailbox
- Folder contents live in one specific mailbox for that folder
  - All content operations are redirected to the mailbox for that folder
- All Public Folder mailboxes listen for hierarchy changes and update similar to Outlook clients
  - Folder hierarchy changes are intercepted and written to writeable copy of Public Folder hierarchy
- When a Public Folder mailbox reaches desired size, move some folders/content to new mailbox

Modern Public Folders

- Determine if public folder mailbox has up-to-date copy of hierarchy

```
Get-PublicFolder \ -recurse -resultsize unlimited | where {$_.contentmailboxname -eq "<PF Mailbox ID>"}
```

- List only public folders with content using new -ResidentFolders parameter in CU1

```
Get-PublicFolder \ -recurse -resultsize unlimitated -Mailbox <PF Mailbox ID> -
ResidentFolders
```

Modern Public Folders

- Move public folder content between mailboxes

```
New-PublicFolderMoveRequest -Folders "\Folder1","\Folder1\Child1" -TargetMailbox <PF Mailbox Identity>
```

- Notes:
  - Subfolders underneath specified folders are not moved by default
  - Each folder to be move must be listed in the –Folders parameter
  - Only one move request at a time; use Remove-PublicFolderMoveRequest to remove completed move requests

Thank You!

Microsoft

Scott Schnoll
Principal Technical Writer
scott.schnoll@microsoft.com
http://aka.ms/schnoll

schnoll

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.