UTORexchange & Office 365

# Developing a Migration Planning Tool

# What is Migplanner

- Who's Used It?
- Department Facing Features
  - Group Definition
  - Data Remediation
  - Scheduling
  - High level progress updates
- Project Team Features
  - Detailed Progress Analysis
  - Searchable Data with Context
  - Migration Group Issues Surveillance

# What is this Talk?

- How the Migplanner came about
- The technology stack Migplanner runs on
- Features of Migplanner you might not have encountered

# My Background

- U of T Alumni
  - BA 2011 – Ethics, Society, & Law; Bioethics; Political Science
  - MI 2013 – Knowledge Management & Information Management
- History with the Email Project
  - User Support, Documentation, Testing for Live@EDU
  - Testing, Documentation, Research & Writing for O365 conversion and expansion to Faculty/Staff
- History with U of T IT
  - Assistant Help Desk Advisor for Information Commons Help Desk
  - David Sutherland

The Phantom Solution

# Episode I

# The Migration Problem

- Mesh of collaboration
- Disruption to the organization
- Data for insights
- A queryable model of our infrastructure
- We were looking for VIP anchored clusters

Attack of the Feature Requests

# Episode II

# Pilot Migrations

Migration System

- Dell/Quest Tools:
  - Mail Migrator for Exchange (MMEX)
  - Mail Migrator for Active Directory (MMAD)
- Collection based control system
- Coordinated actions in multiple systems

Migration Coordination

- Emailed Excel Spreadsheets
- Aggressive rollout by distributed team, separated from migration execution and support
- Need improved coordination

# Feature Growth

- Priorities Tracked and Coordinated through our Trello System
- Summary
  - 52 Trello features
  - 198 Commits on Backend
  - 134 Commits on Frontend

- Highlights to Minimum Viable
  - Produce a file to feed control scripts to automate parts of the migration process
  - Group Building & Scheduling
  - Migration Progress Tracking

# The Technology

# Data Storage: Neo4j

- Neo4j the Company
  - Swedish founded and GPL open sourced Neo4j in 2007
  - Now US based series D funded
  - https://neo4j.com


- Neo4j the Product
  - Used in Finance for fraud detection
  - Used in Retail for Customer Insight
  - Enterprise Licence for High Availability and Performance
  - Community Licence under GPL is fully featured but some limits on size
  - Official Drivers: Java, JavaScript, Python, .NET

# Data Storage: Neo4j

- Graph Databases Overview

# Data Storage: Neo4j

- Graph Databases Overview
- Data Description
  - Nodes
  - Edges
  - Labels
  - Properties

# Data Storage: Neo4j

- Graph Databases Overview
- Data Description
  - Nodes
  - Edges
  - Labels
  - Properties
- Analogies to RDBMS
  - “Primary Keys”
  - Indexes

# MigPlanner's Neo4j Schema

Nodes

- User
- Resource
- Department

Edges

- HasAccess
- Owner
- FullAccess
- MemberOf
- DeptPartOf

# MigPlanner's Neo4j Schema

Nodes

- User
  - Name
  - UserId*
  - SrcSam
  - Domain
  - MbxSize
  - smbx
  - faGroup
  - saGroup
  - Detail
- Resource
- Department

Edges

- HasAccess
  - Authorization
  - Target
- Owner
- FullAccess
- MemberOf
- DeptPartOf

# MigPlanner's Neo4j Schema

Nodes

- User
- Resource
  - Name
  - Alias*
  - SrcAlias
  - Domain
  - Type
  - Email
  - MbxSize
- Department

Edges

- HasAccess
- Owner
- FullAccess
- MemberOf
- DeptPartOf

# MigPlanner's Neo4j Schema

Nodes

- User
- Resource
- Department
  - Name
  - ShortCode*
  - FisCode

Edges

- HasAccess
- Owner
- FullAccess
- MemberOf
- DeptPartOf

# MigPlanner's Neo4j Schema

Nodes

- User
- Resource
- Department
  - Name
  - ShortCode*
  - FisCode

Edges

- HasAccess
- Owner
- FullAccess
- MemberOf
- DeptPartOf

```
UOFT.GOVCN.PRES.VPUO.CIO
UOFT.GOVCN.PRES.VPUO.CIO.ACT
```

# Neo4j Data Manipulation

- Cypher Query Language
- Basic Query Structure

```
1 MATCH (Node)-[Edge]-(Node), (Node)
2 WHERE Conditions
3 RETURN Node.Property
```

# Data Storage: Neo4J

- Data Manipulation: Cypher
- Basic Query Structure
- Simple Queries

```
MATCH (u:User)-[:MemberOf]->(d:Department)
WHERE (
        d.ShortCode = 'UOFT.GOVCN.PRES.VPUO.CIO'
        OR d.ShortCode = 'UOFT.GOVCN.PRES.VPUO.CIO.ACT'
    )
    AND u.Domain IN ['UTORmail','UTORexchange']
    AND u.smbx = "false"
RETURN u.UserId
```

```
MATCH (u1:User),(u2:User)
WHERE u1.UserId = 'suther43'
    AND u2.UserId = 'helpdesk'
CREATE (u1)-[r:FullAccess]->(u2)
RETURN *
```

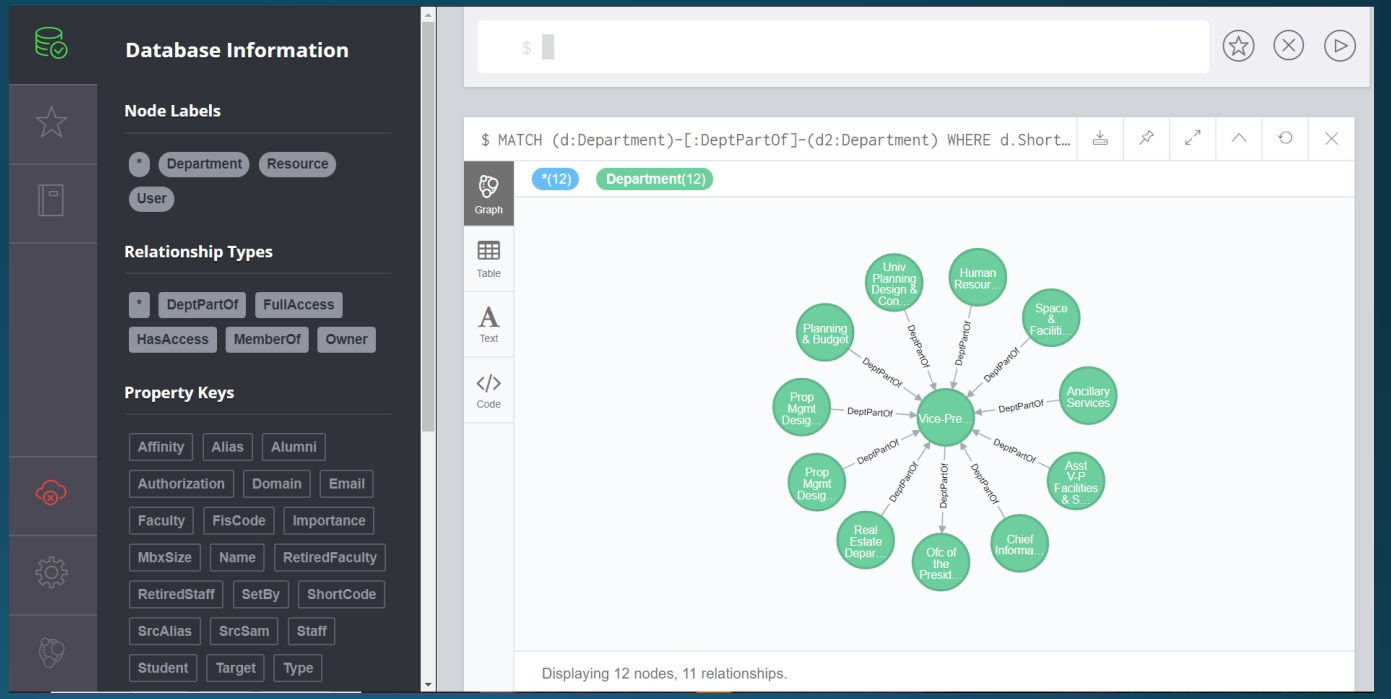
# Neo4j Data Manipulation

- Data Manipulation: Cypher
- Basic Query Structure
- Simple Queries
- Complex Queries

```
MATCH (mig:User)
WHERE mig.UserId = 'suther43' OR mig.UserId = 'zhaonina'
WITH mig
MATCH (mig)-[:Owner|Manager]-(smbx:User)
WITH smbx
MATCH (Ext:User)-[:Owner|Manager]-(smbx)
WHERE Ext.UserId <> 'suther43' AND Ext.UserId <> 'zhaonina'
OPTIONAL MATCH (Ext)-[:MemberOf]->(d2:Department)
RETURN Ext.UserId, Ext.Name, d2.ShortCode, d2.Name
ORDER BY d2.ShortCode
```

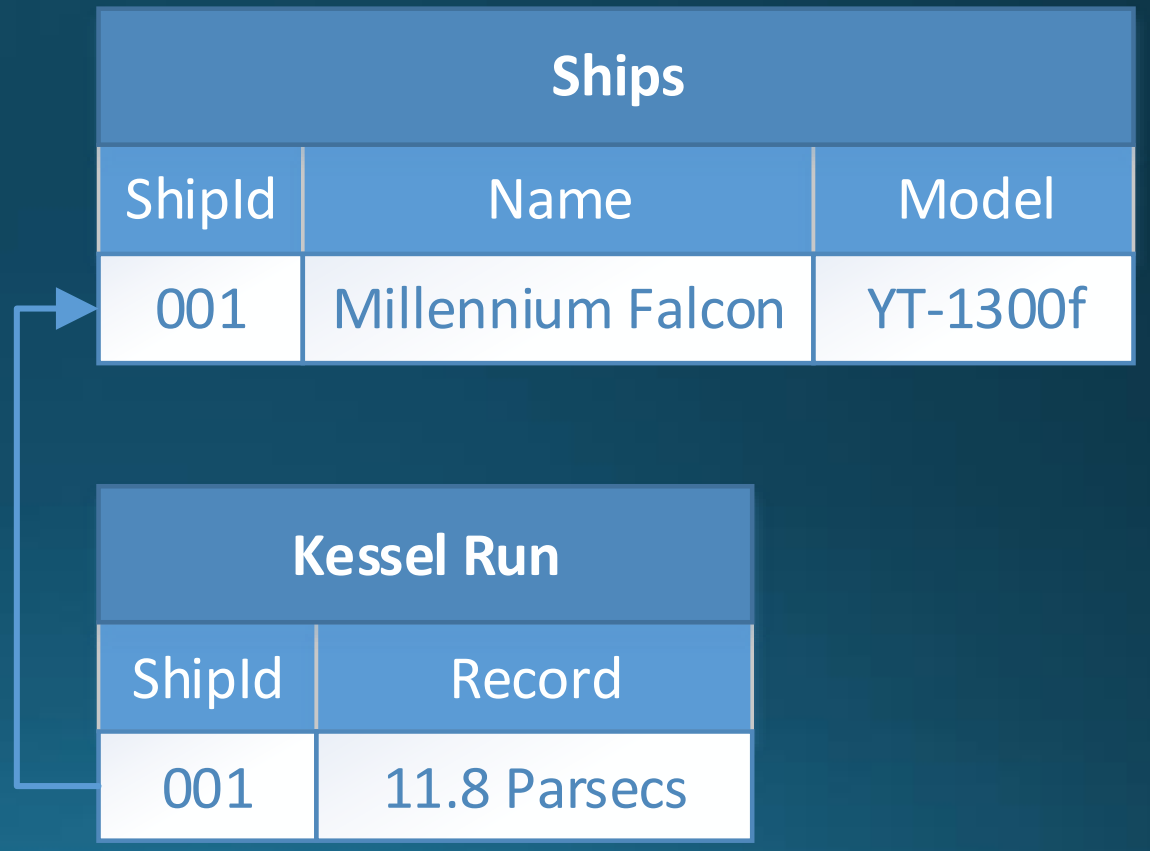
# Data Storage: Neo4J

- Web console
- Operates under JVM
- Controls through HTTP and BOLT

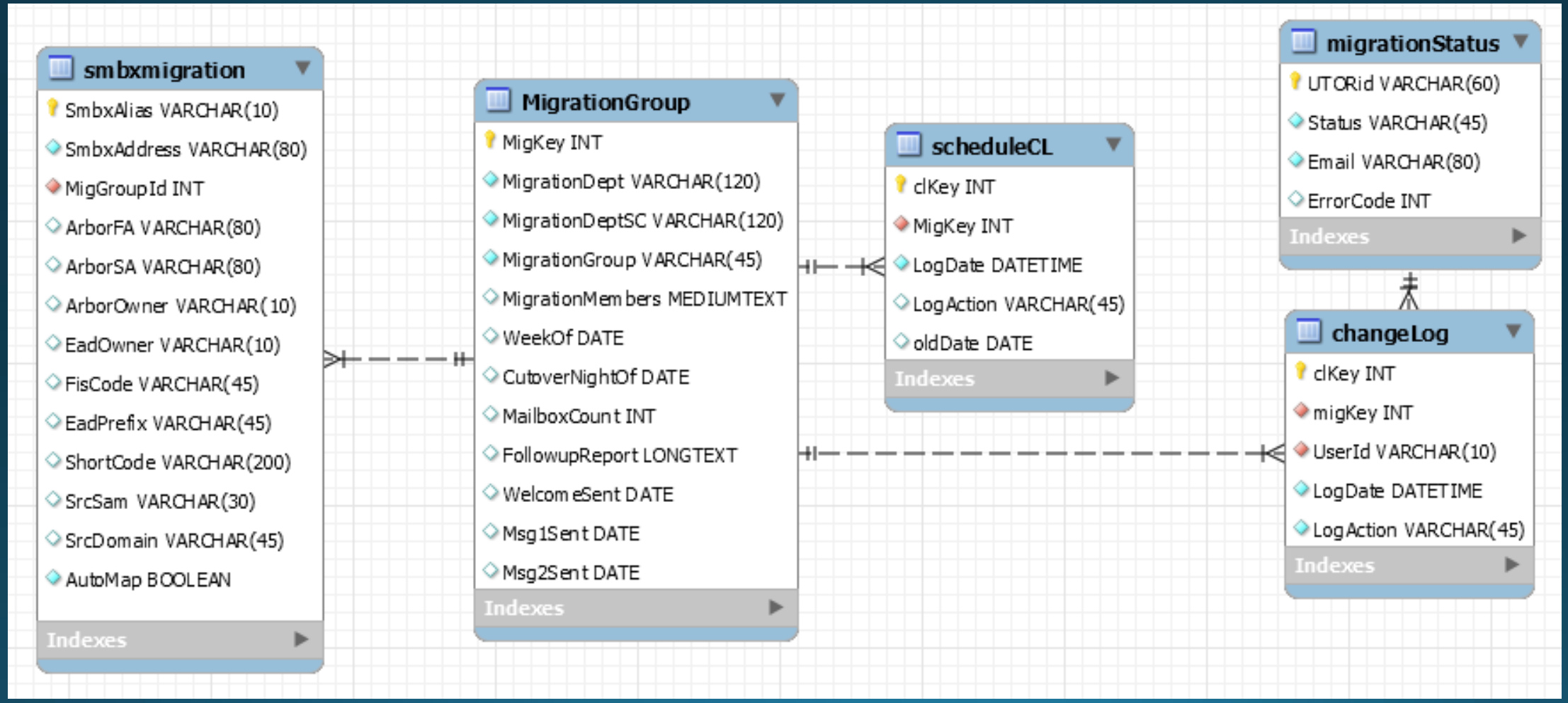


# Data Storage: MySQL

- Persistent Data v. Volatile Data
- Why MySQL?
- Data Description
  - Tables
  - Foreign Keys
- Migplanner Implementation
  - Migration Planning, Tracking
  - Authorization
  - Infrastructure Data
  - Data Remediation

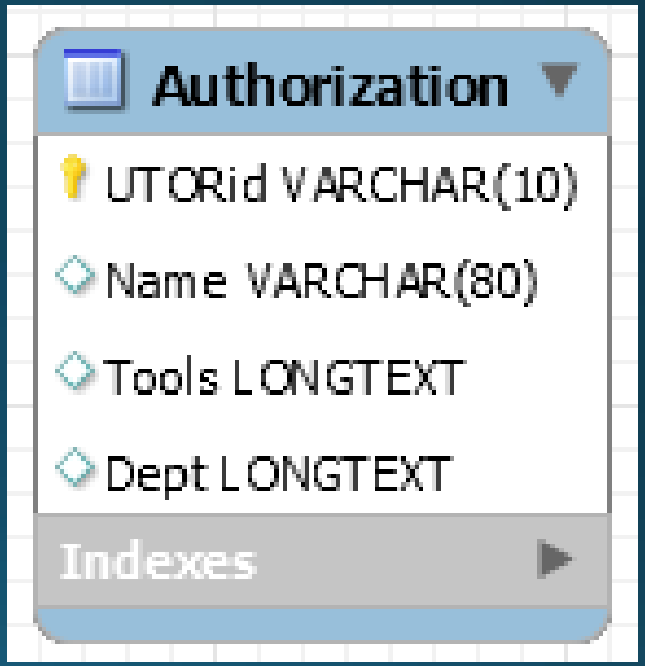


# Migplanner MySQL Schema: Migration Planning & Tracking

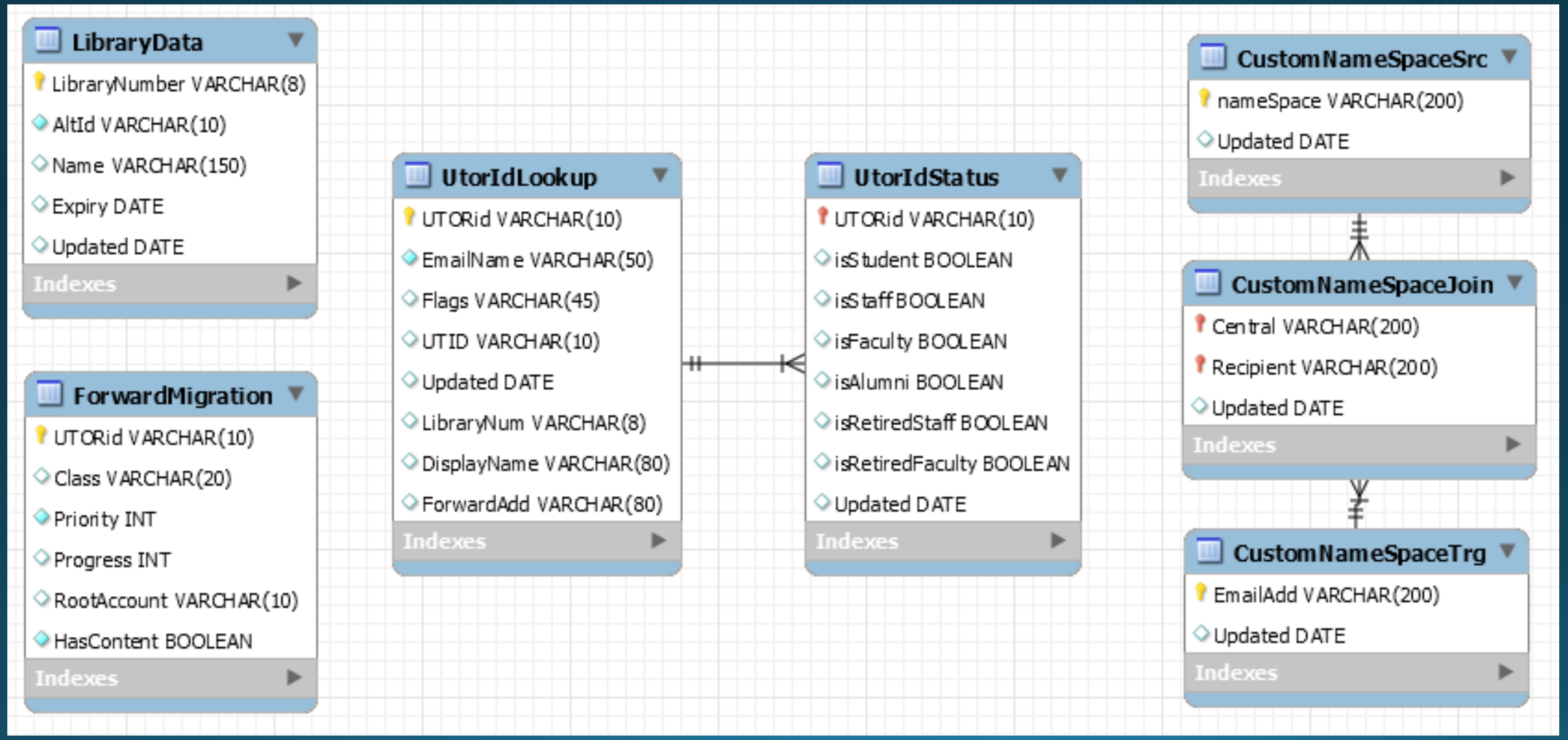


# Migplanner MySQL Schema: Authorization

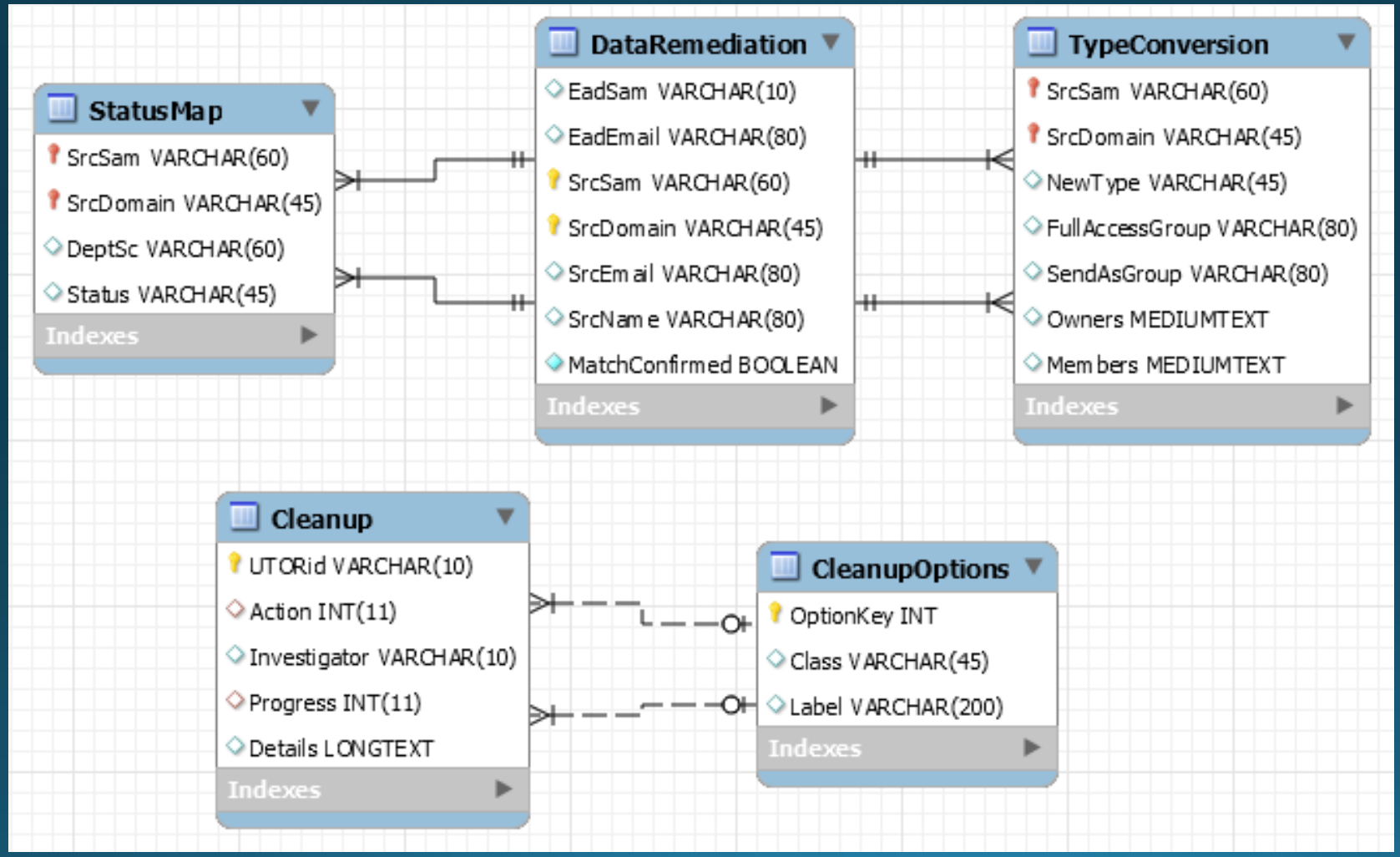
- Tools: JSON structured
  - ScheduleOnly
  - DeptView
  - DeptAdmin
  - DeptSchedAdmin
  - ProjAdmin
  - FullAdmin
- Dept: JSON array of ShortCode



# Migplanner MySQL Schema: Infrastructure Data



# Migplanner MySQL Schema: Data Remediation



# Data Storage: MySQL

- Query Structure

```
1  SELECT Columns
2  FROM Tables
3  WHERE Condition
```

# Data Storage: MySQL

- Query Structure
- Notable Queries

```sql
SELECT  dr.*, sm.DeptSc, sm.Status, tc.NewType, tc.Owners, tc.Members
FROM DataRemediation dr
LEFT JOIN StatusMap sm ON sm.SrcDomain = dr.SrcDomain AND sm.SrcSam = dr.SrcSam
LEFT JOIN TypeConversion tc ON tc.SrcDomain = dr.SrcDomain AND tc.SrcSam = dr.SrcSam
WHERE dr.SrcDomain = :d AND dr.EadSam IS NOT NULL AND dr.EadEmail IS NOT NULL
```

```sql
SELECT UTORid, Flags
FROM UtorIdLookup
WHERE BINARY(Flags) RLIKE '[mnOgC]' AND UTORid IN (
    SELECT EadSam
    FROM DataRemediation
    WHERE EadSam IS NOT NULL
)
```

# Hosting: ITS - EIS

- Data Center based Virtual Machines
- Architecture
  - Frontend
  - Backend
- Frontend: CentOS for Web-login integrated Linux, Apache, PHP
- Backend: Self-administered Ubuntu for a Docker Host

# Hosting: Docker

- Docker the Company
  - Launches in 2013
  - Series D funding in 2015
  - Cloud hosted services, data center services
  - Open source Moby Project
- Docker the Product
  - Virtualization through “Containers”
  - Shared kernel for Linux operating systems
  - Library of containers
  - Customizable through Docker builds

# Hosting: Docker

- Why? Solution to change management

```
 3 FROM ubuntu:17.04
22 RUN apt-get update && apt-get install -y \
23     apt-utils \
24     ca-certificates\
25     openssl \
26     apache2 \
27     php \
28     php7.0-curl \
29     php-bcmath \
30     php7.0-mbstring \
31     libapache2-mod-php \
32     composer \
33     wget \
34     unzip \
35     curl \
36     mysql-server \
37     phpmyadmin \
38     cron \
39     nano
```

# Hosting: Docker

- Why? Solution to change management
- Data Persistence Soliton
  - Databases
  - File Caches
  - Logs
- UTORrecover Backups
- Build scripts

```
5  docker run -i \
6      -v /home/DockerData/Neo4j:/var/lib/neo4j/data/databases \
7      -v /home/DockerLog/Neo4j:/var/log/neo4j \
8      -v /home/DockerData/MySql:/var/lib/mysql \
9      -v /home/DockerData/MySqlArchive:/home/MySqlArchive \
10     -v /home/DockerLog/MySql:/var/log/mysql \
11     -v /home/DockerLog/Apache:/var/log/apache2 \
12     -p 80:80 -p 443:443 -p 7474:7474 -p 7473:7473 -p 7687:7687 -p 1337:1337 \
```

# Application: PHP

- Choosing PHP
- Neo4J Driver
- Custom API
- Object Inclined

- Front End
  - 54 files, 1 Class (User)
  - User Class: 7 Methods, 137 lines
- Back End
  - 34 files, 1 Class (Database)
  - Database Class: 32 Methods, 96 Static Functions. 3,463 lines.
  - 90 Endpoints on the API

# Application: Bootstrap

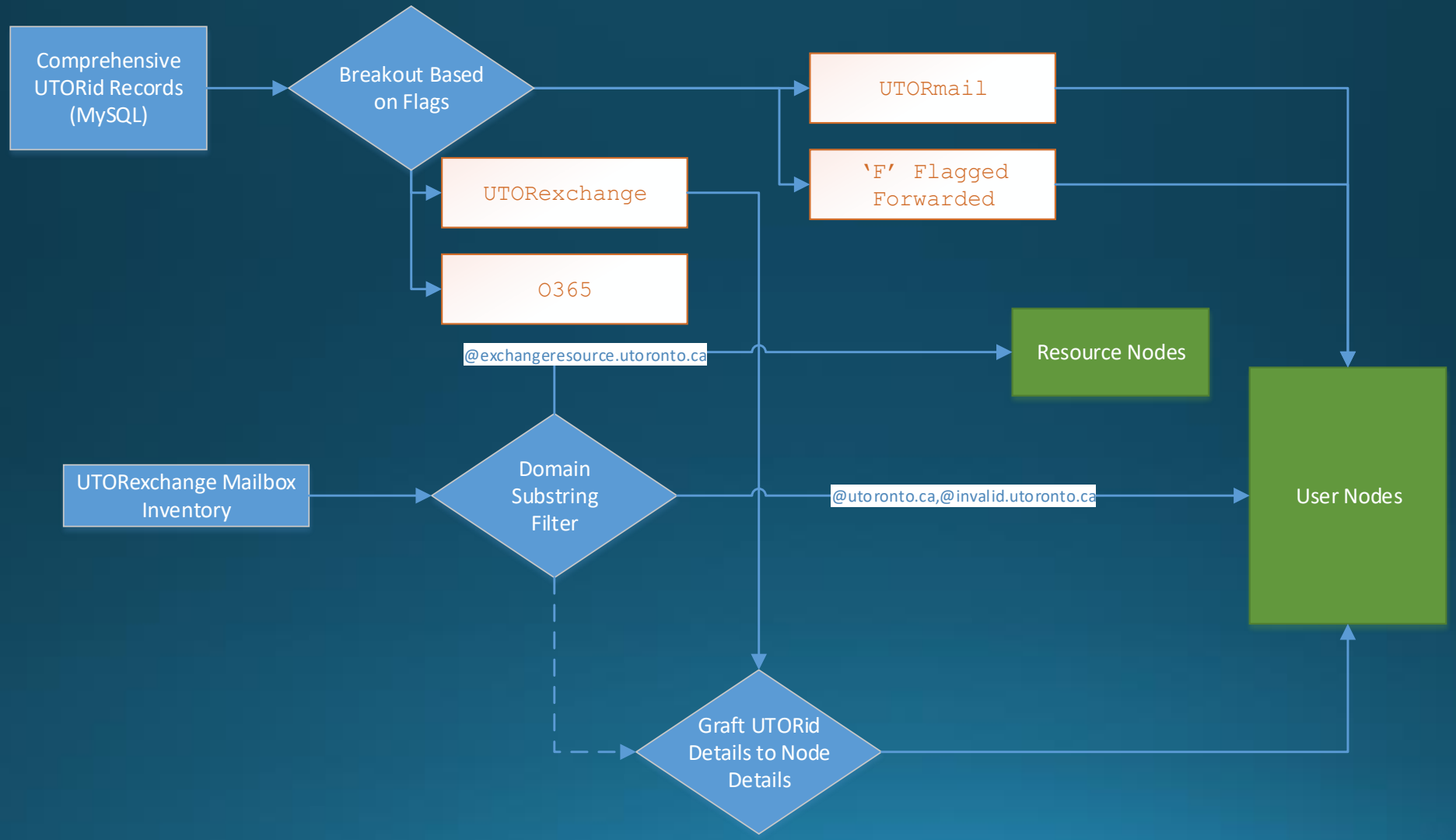
- Frontend framework
- Interface design with little/no thought
- Requires jQuery so jQuery adopted for client side work

# Data Integration

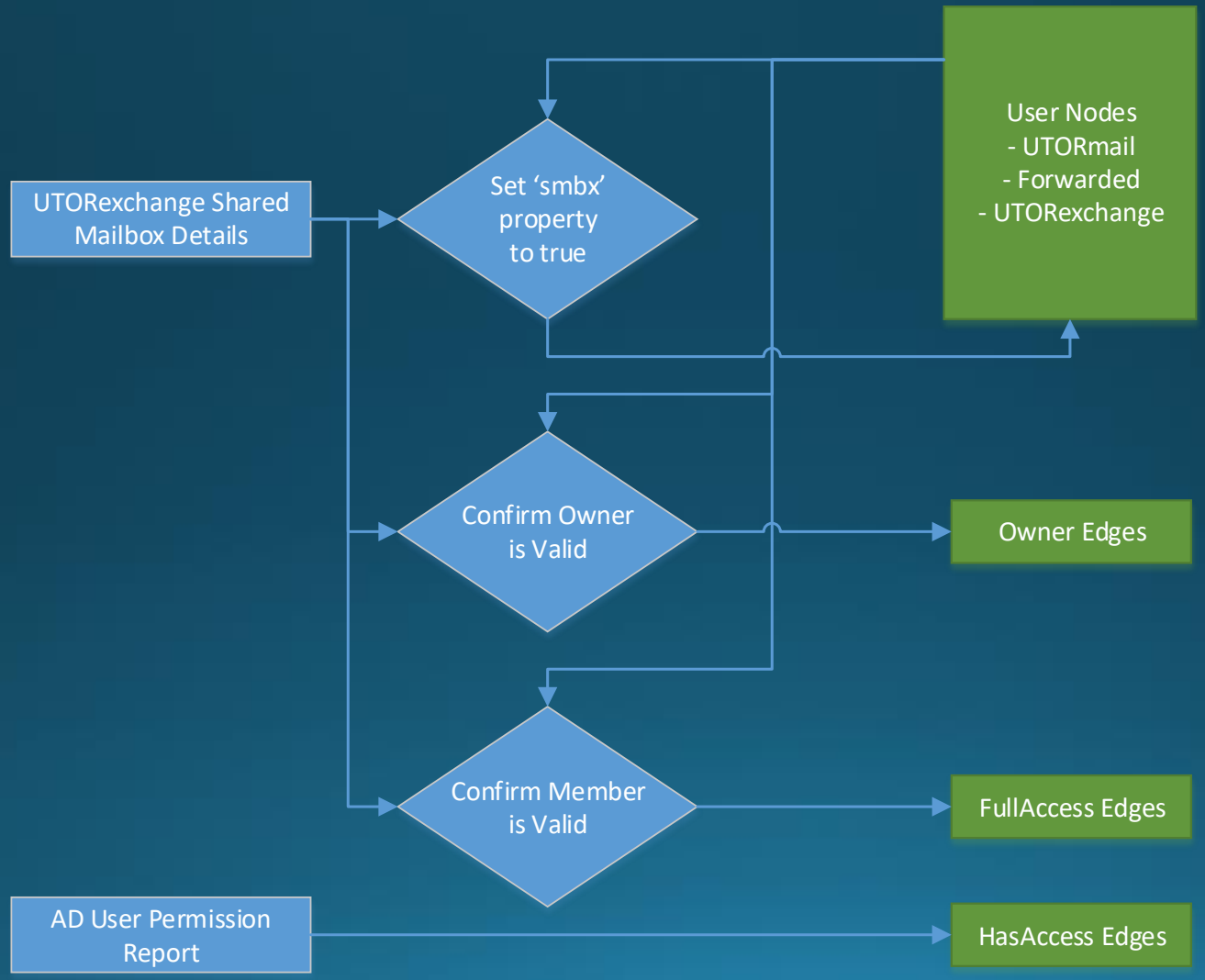
# Data Sources

- Data Files
  - Medicine Exchange Mailbox Inventory
  - UTORexchange Mailbox Inventory
  - UTORexchange Shared Mailbox Group Data
  - UTORexchange inter-account permissions
  - EAD Department Affiliation
  - Department List
  - Other Data Sources User Accounts
- MySQL Data
  - Remediation Data
    - Point-in-time audit of Medicine
    - Manually/Automated Review of SIRSI Mail-only borrowers
  - Comprehensive UTORid Records
    - Periodically updated through Data file.

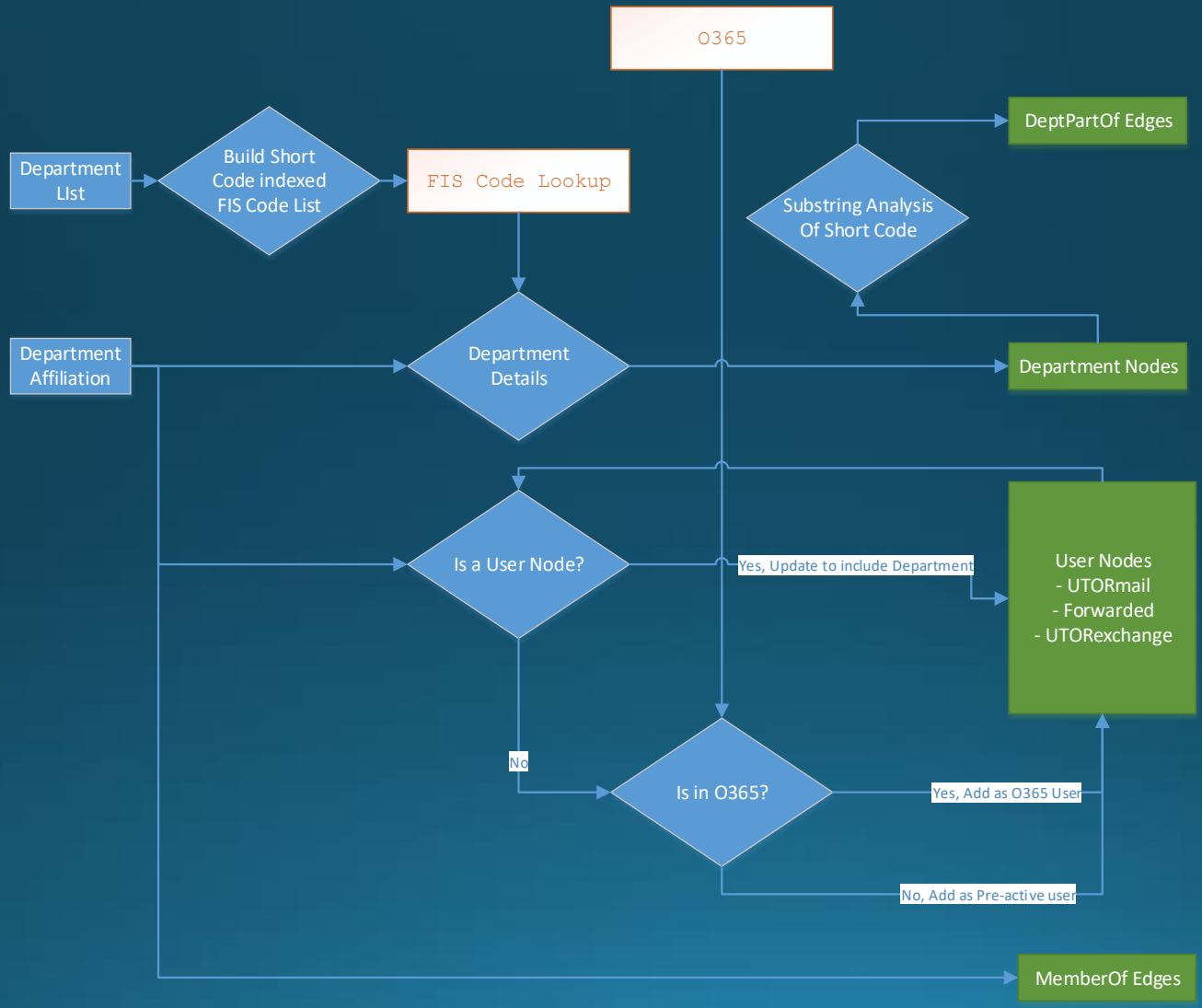
# Integration Process



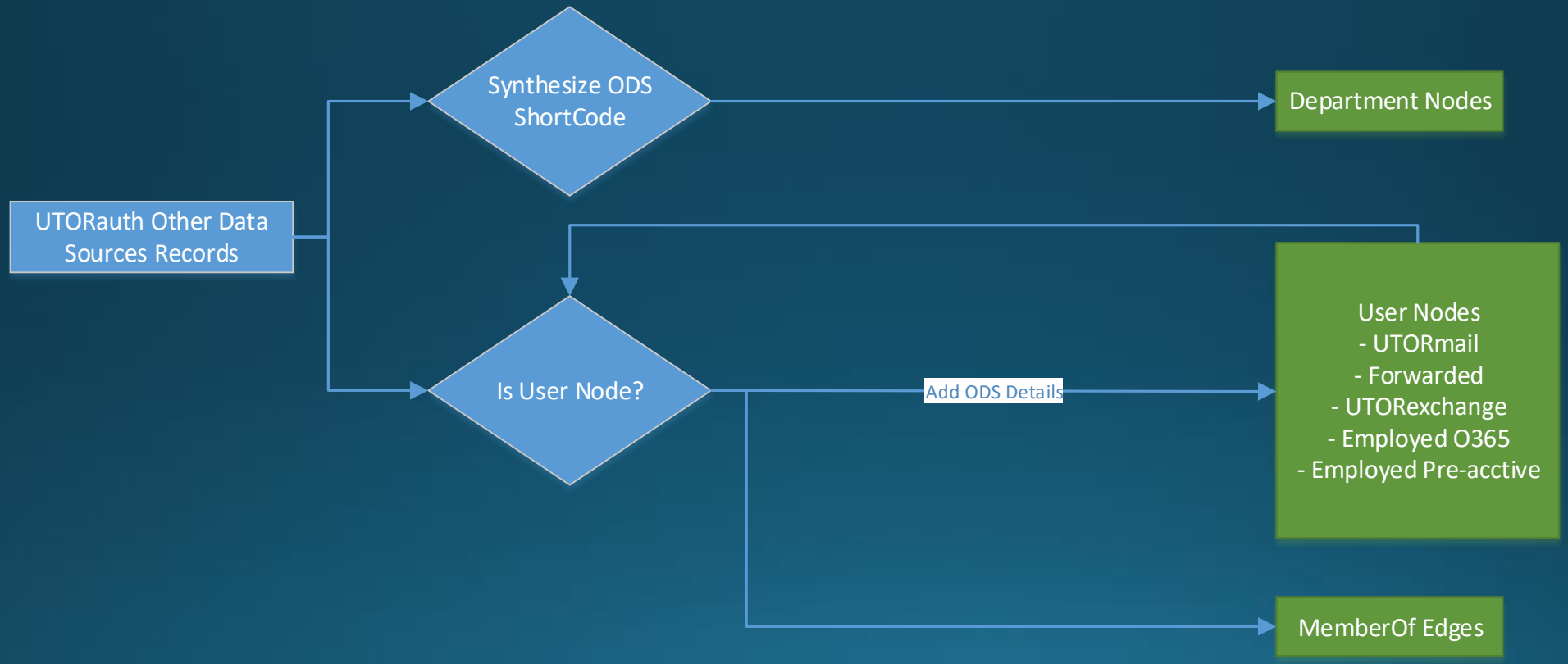
# Integration Process



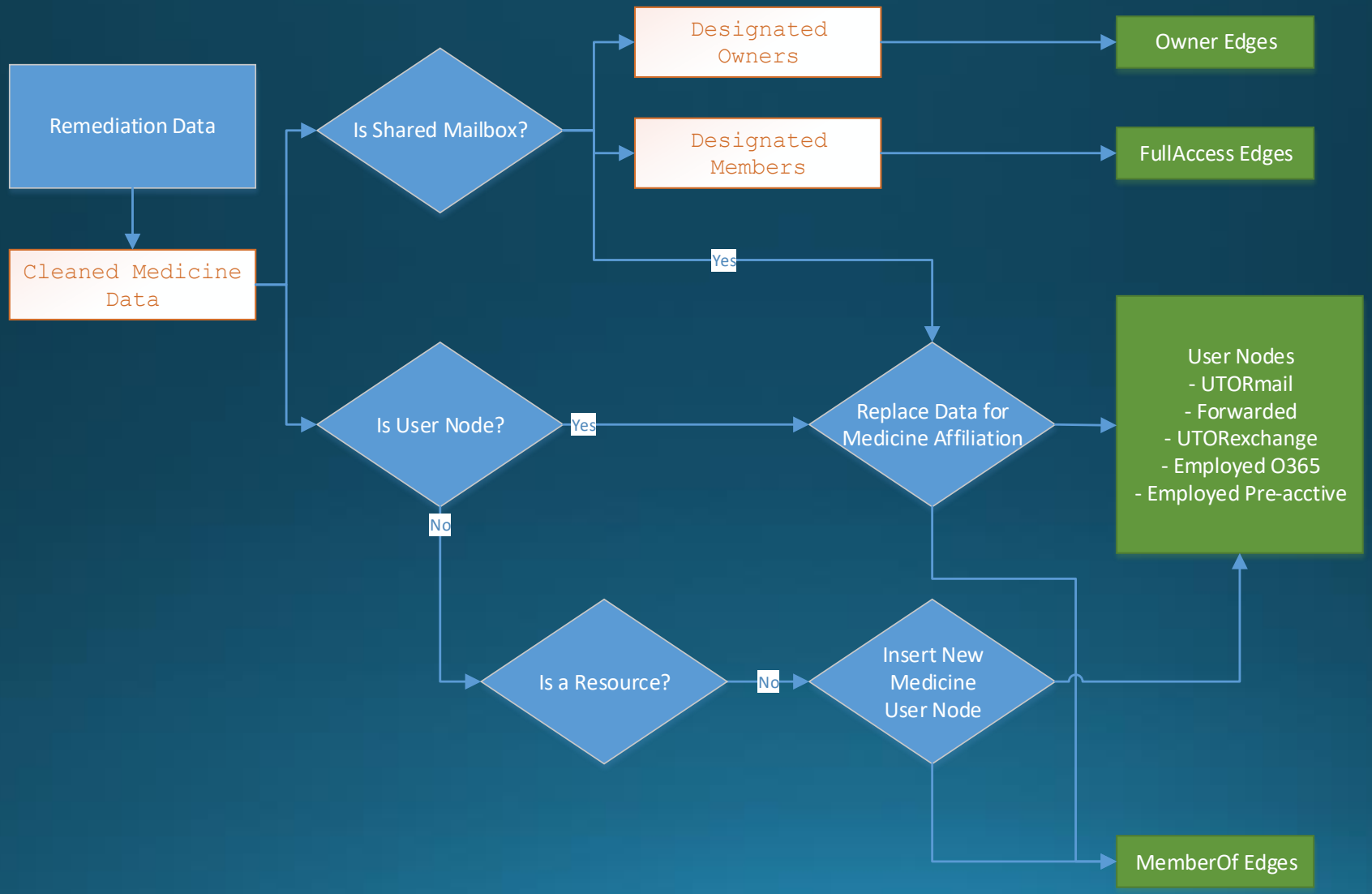
# Integration Process



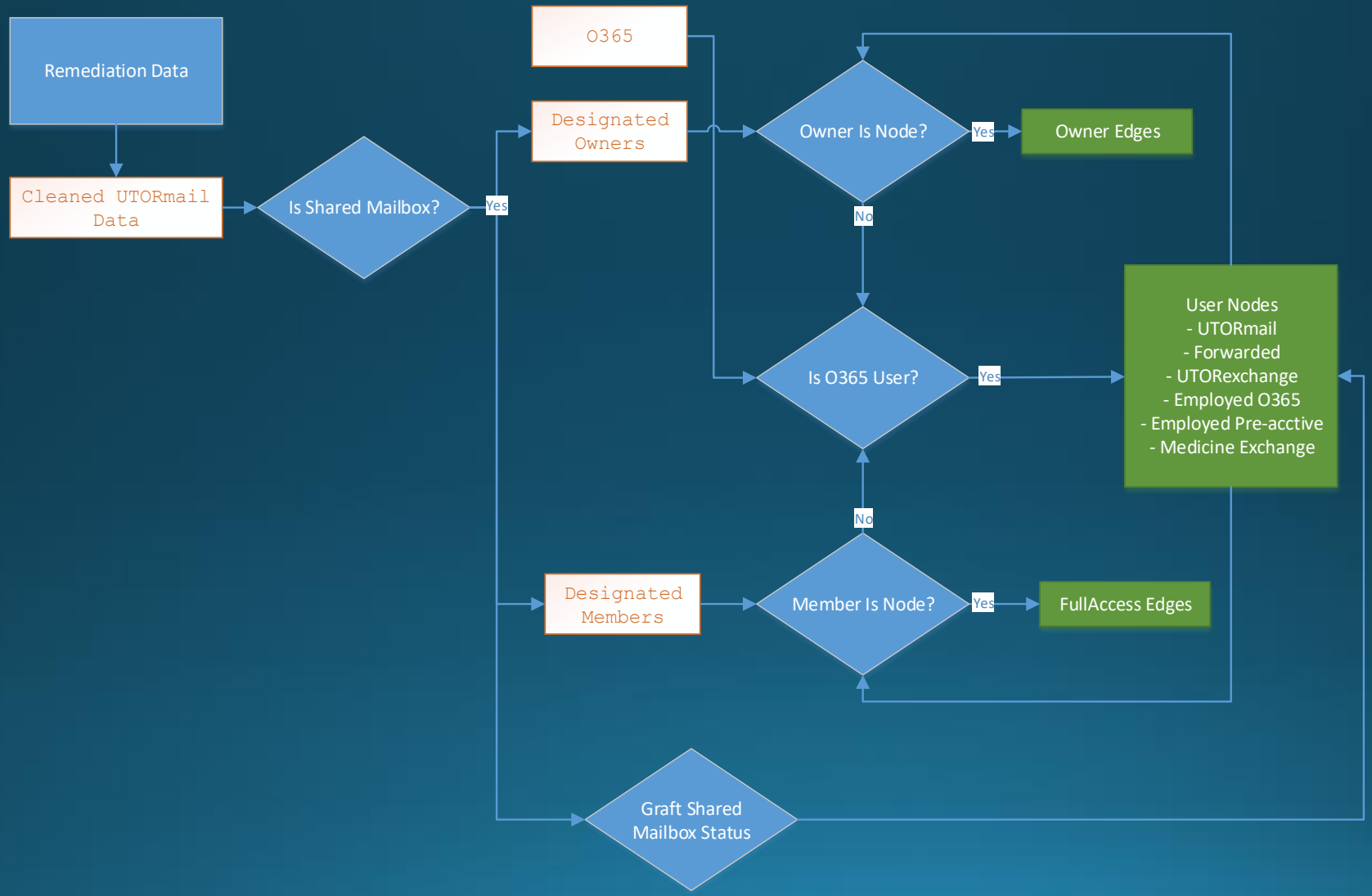
# Integration Process



# Integration Process

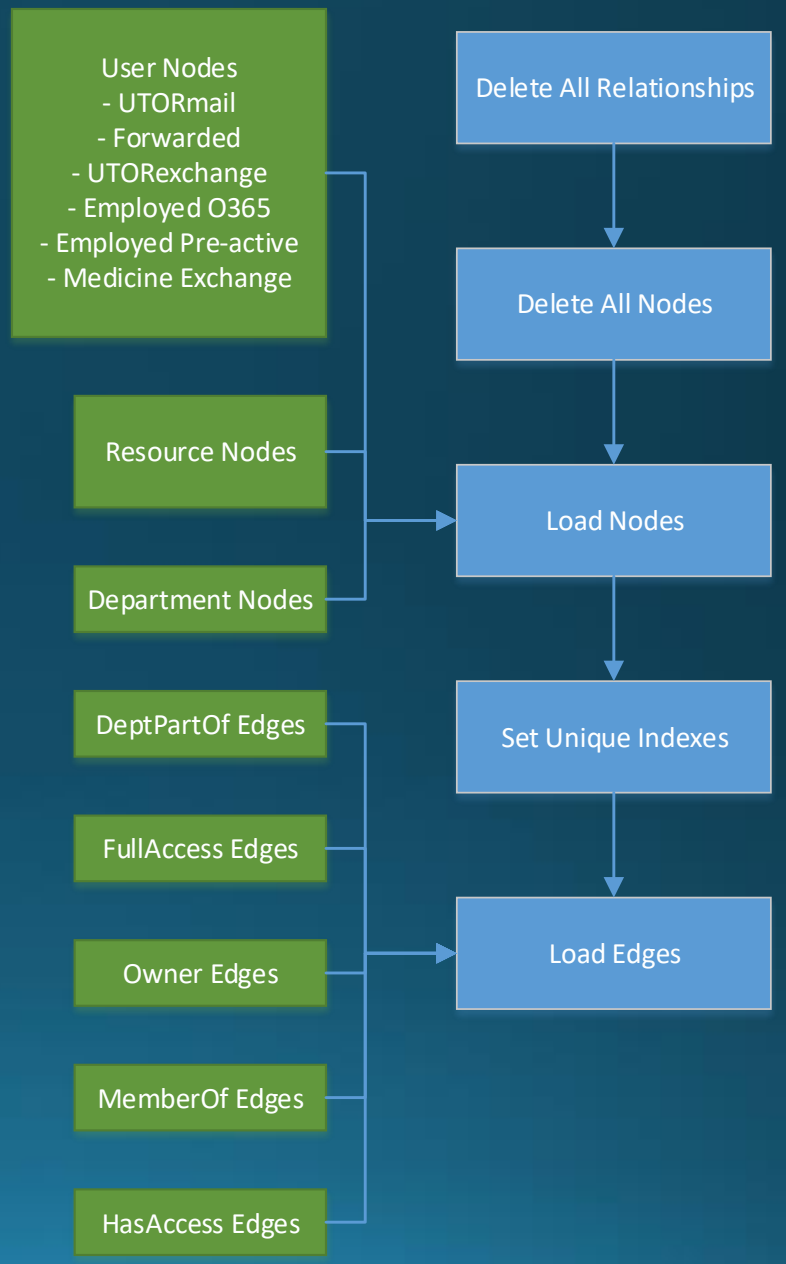


# Integration Process



# Integration Process

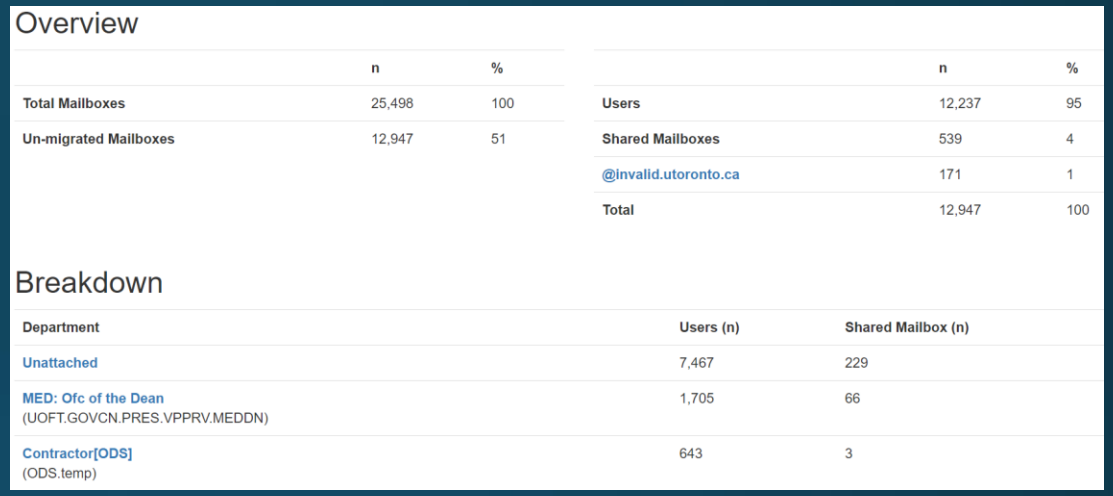
- Run in a separate POST command.
- Leverage Native Neo4j Batch Import from CSV features



# Admin Features

# Migration Control & Progress Tracking

- Overview Covers Medicine Exchange, UTORexchange, UTORmail
- Detailed Analysis covers UTORexchange
- Checks By Week for:
  - Users not ready to be Cutover
  - Users already Cutover
  - Users Forwarding
- Changes Logs of group actions



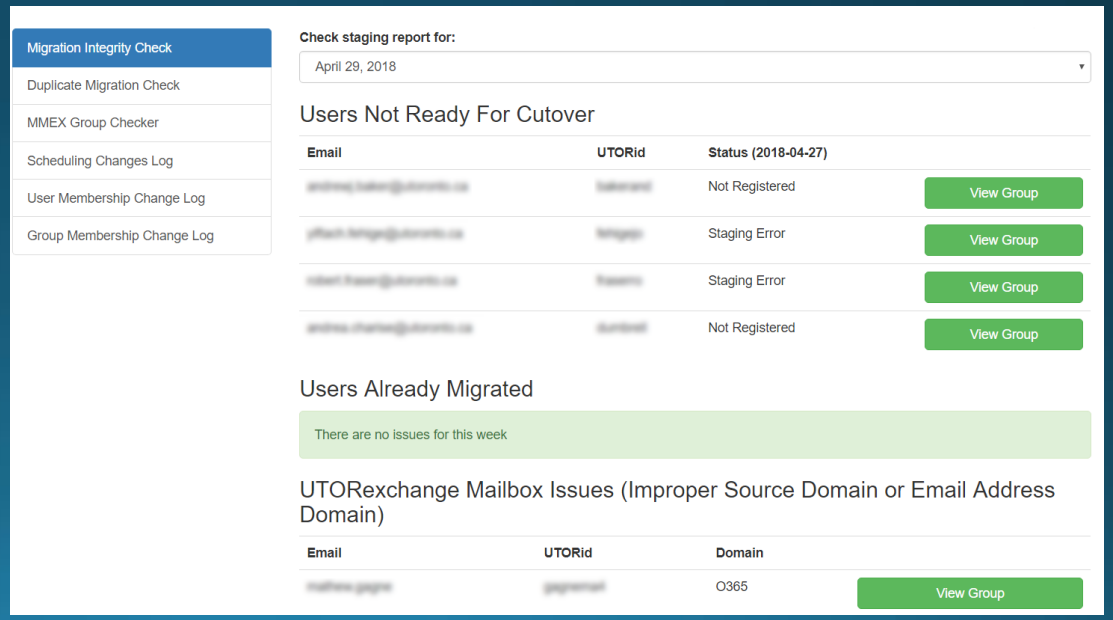
Overview

| | n | % |
|---|---|---|
| Total Mailboxes | 25,498 | 100 |
| Un-migrated Mailboxes | 12,947 | 51 |

| | n | % |
|---|---|---|
| Users | 12,237 | 95 |
| Shared Mailboxes | 539 | 4 |
| @invalid.utoronto.ca | 171 | 1 |
| Total | 12,947 | 100 |

Breakdown

| Department | Users (n) | Shared Mailbox (n) |
|---|---|---|
| Unattached | 7,467 | 229 |
| MED: Ofc of the Dean (UOFT.GOVCN.PRES.VPPRV.MEDDN) | 1,705 | 66 |
| Contractor[ODS] (ODS.temp) | 643 | 3 |



Migration Integrity Check
Duplicate Migration Check
MMEX Group Checker
Scheduling Changes Log
User Membership Change Log
Group Membership Change Log

Check staging report for:
April 29, 2018

Users Not Ready For Cutover

| Email | UTORid | Status (2018-04-27) | |
|---|---|---|---|
| [redacted] | [redacted] | Not Registered | View Group |
| [redacted] | [redacted] | Staging Error | View Group |
| [redacted] | [redacted] | Staging Error | View Group |
| [redacted] | [redacted] | Not Registered | View Group |

Users Already Migrated

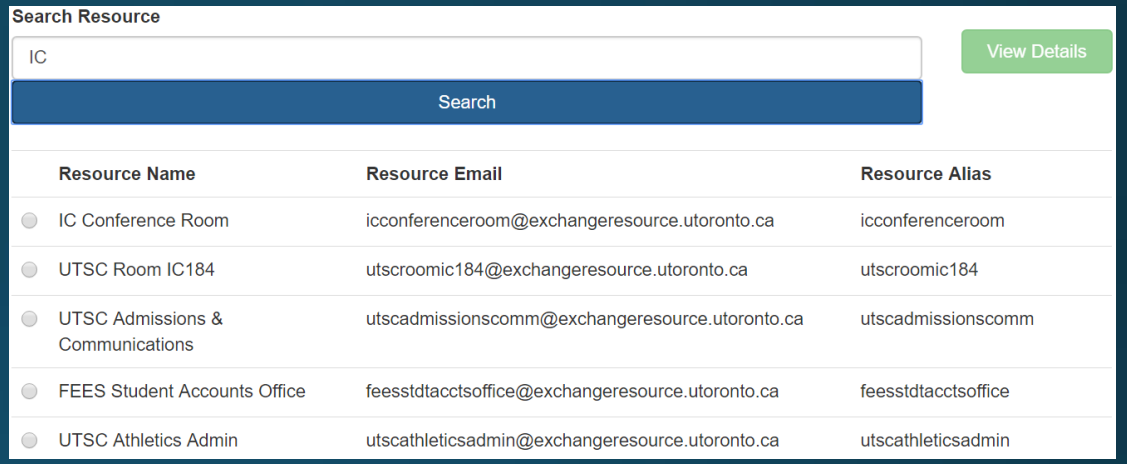
There are no issues for this week

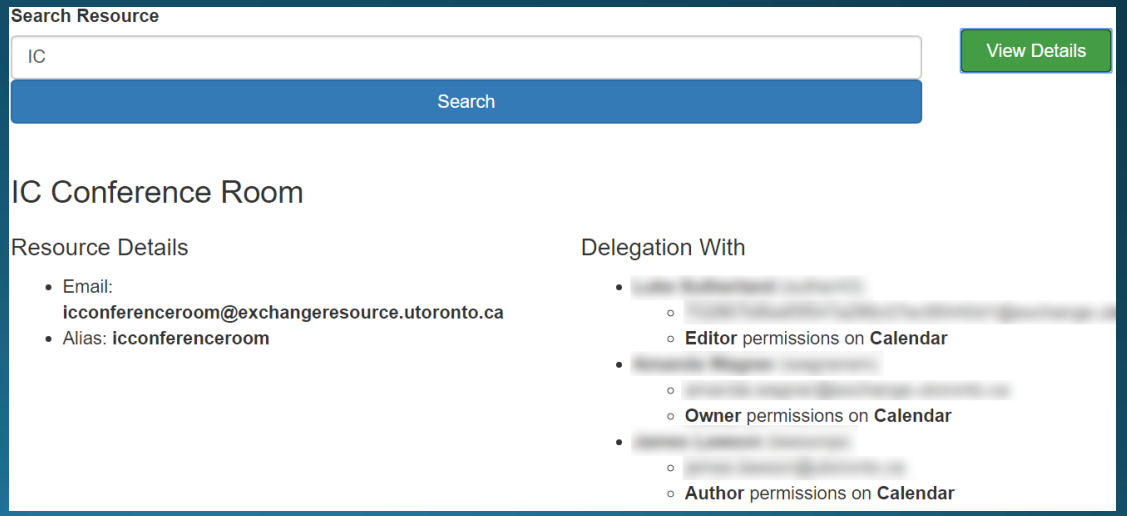
UTORexchange Mailbox Issues (Improper Source Domain or Email Address Domain)

| Email | UTORid | Domain | |
|---|---|---|---|
| [redacted] | [redacted] | O365 | View Group |

# Resource Search

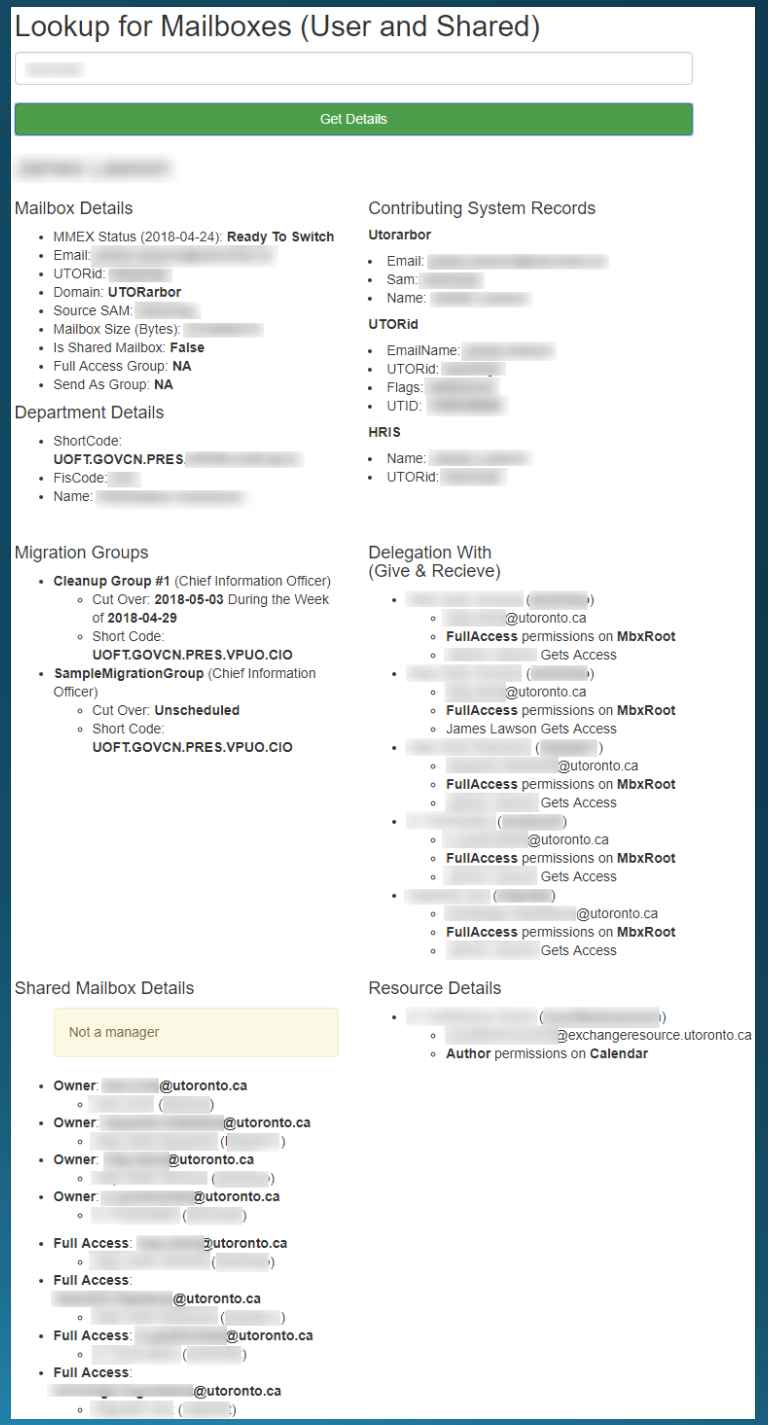
- Substring Search of Resource Name
- Details of the resource



# Mailbox Search

- Search Accepts UTORid or email address (not email name)
- Reports
  - User Node Properties
  - Contributing Systems of Record
  - Department Affiliation
  - Migration Groups
  - Delegation with Accounts
  - Access & Control of Shared Mailboxes



# Questions?

May the Fourth be with you

# Thank You