



USC'S HOME GROWN IDENTITY DATA HUB

SESSION 36044
Thu, Nov 10, 2016
(1:30 PM – 2:15 PM)

PRESENTER

Jay Mathew

Senior Business Systems Analyst

University of the Sunshine Coast

jmathew@usc.edu.au



UNIVERSITY OF THE SUNSHINE COAST

Student enrolments: 11,602

Staff: 906

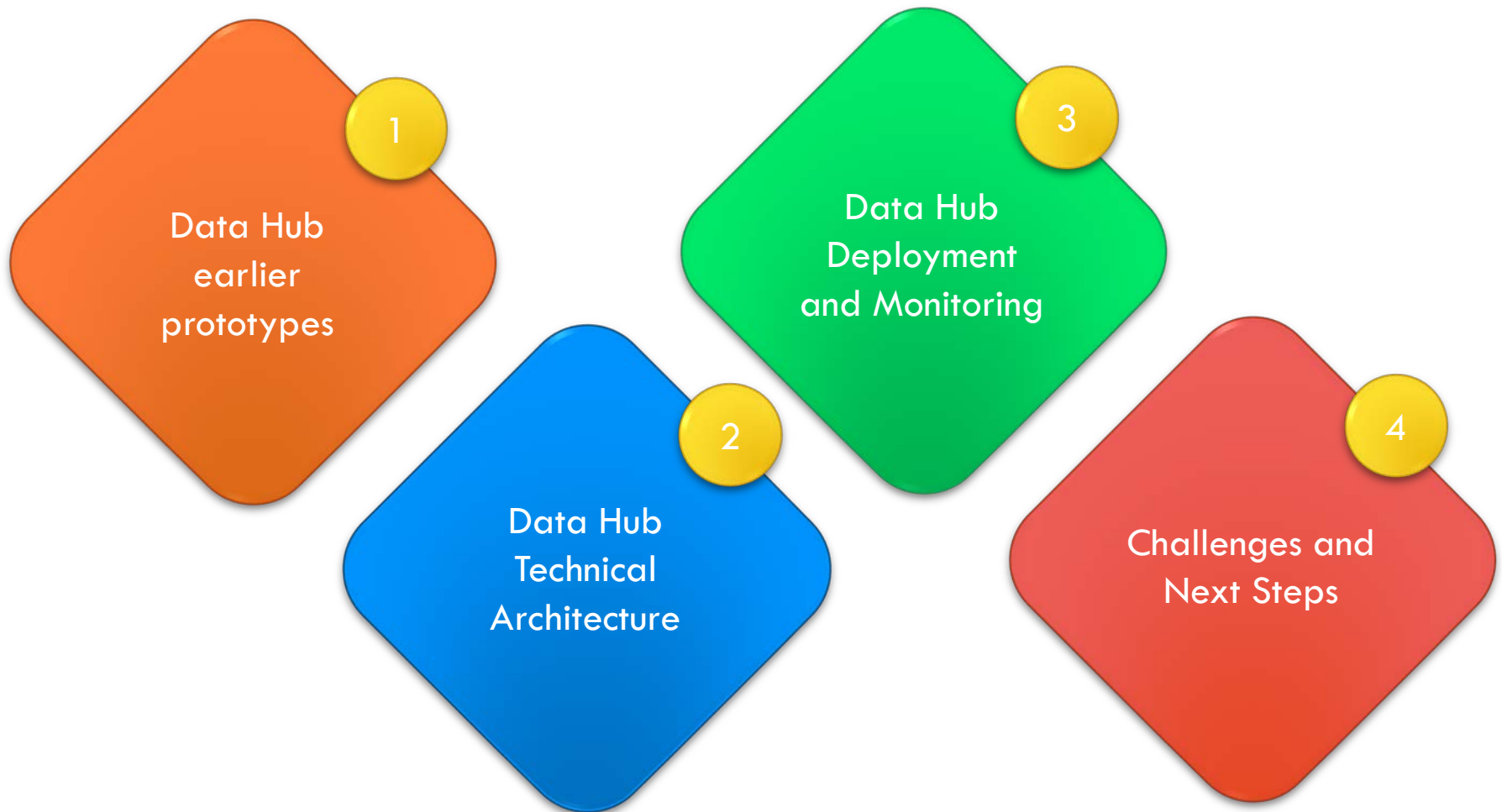
ADU 9-11 NOVEMBER 2016



PEOPLESOFT

- PeopleSoft HCM 9.0 & Campus Solutions 9.0
- PeopleTools 8.54.07
- Windows Server 2012 R2
- Microsoft SQL Server 2014

OVERVIEW





**SORRY
TRY AGAIN**

DATA HUB EARLIER PROTOTYPES

Learnings from earlier
attempts

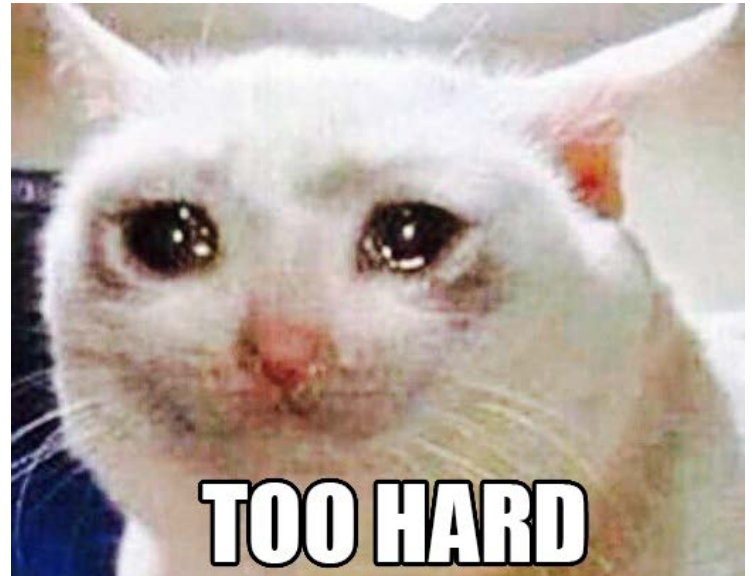
PROTOTYPE #1

- Create provisioning views in both PeopleSoft HCM and CS
- To be queried every 2-5 minutes by Identity Management solution



PROTOTYPE #2

- Fire messages to the Data Hub via Integration Broker
 - Impact analysis required
 - Need to fire IB messages via PeopleCode on save events
 - Need to fire messages on SQR and COBOL person detail updates. Messages wouldn't be real-time.
 - Requires ongoing re-evaluation every time a new PI is released





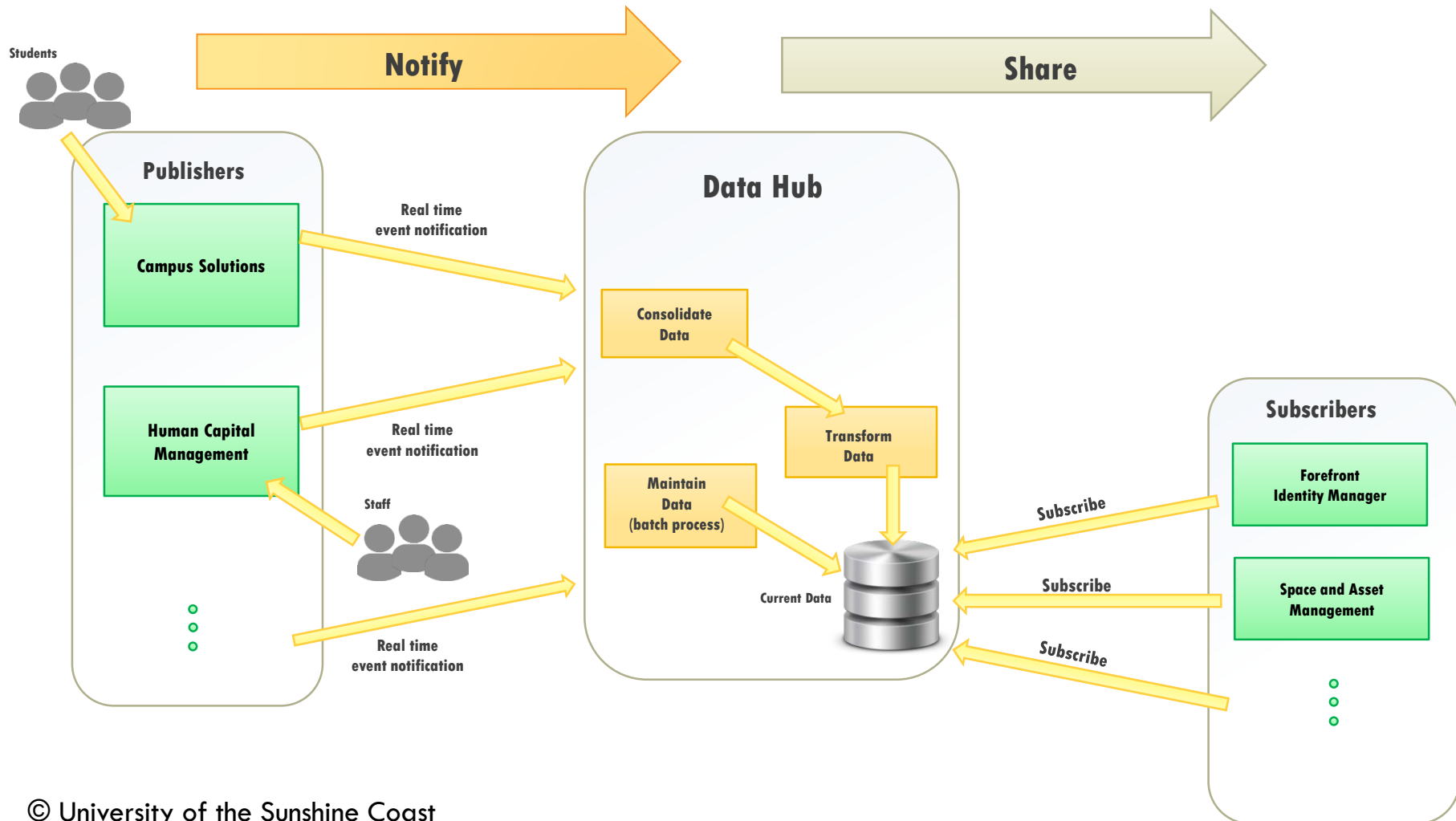
DATA HUB FINAL CONCEPT

A technical overview

DATA HUB FINAL CONCEPT

- Centralised database (running on MS SQL Server 2014) that hosts person identity data for subscribing systems.
- Identity data is sourced from both PeopleSoft and non-PeopleSoft systems
- Data is updated in near real-time using SQL Server technologies.

DATA HUB INTERACTION



DATA HUB REAL-TIME TECHNOLOGIES

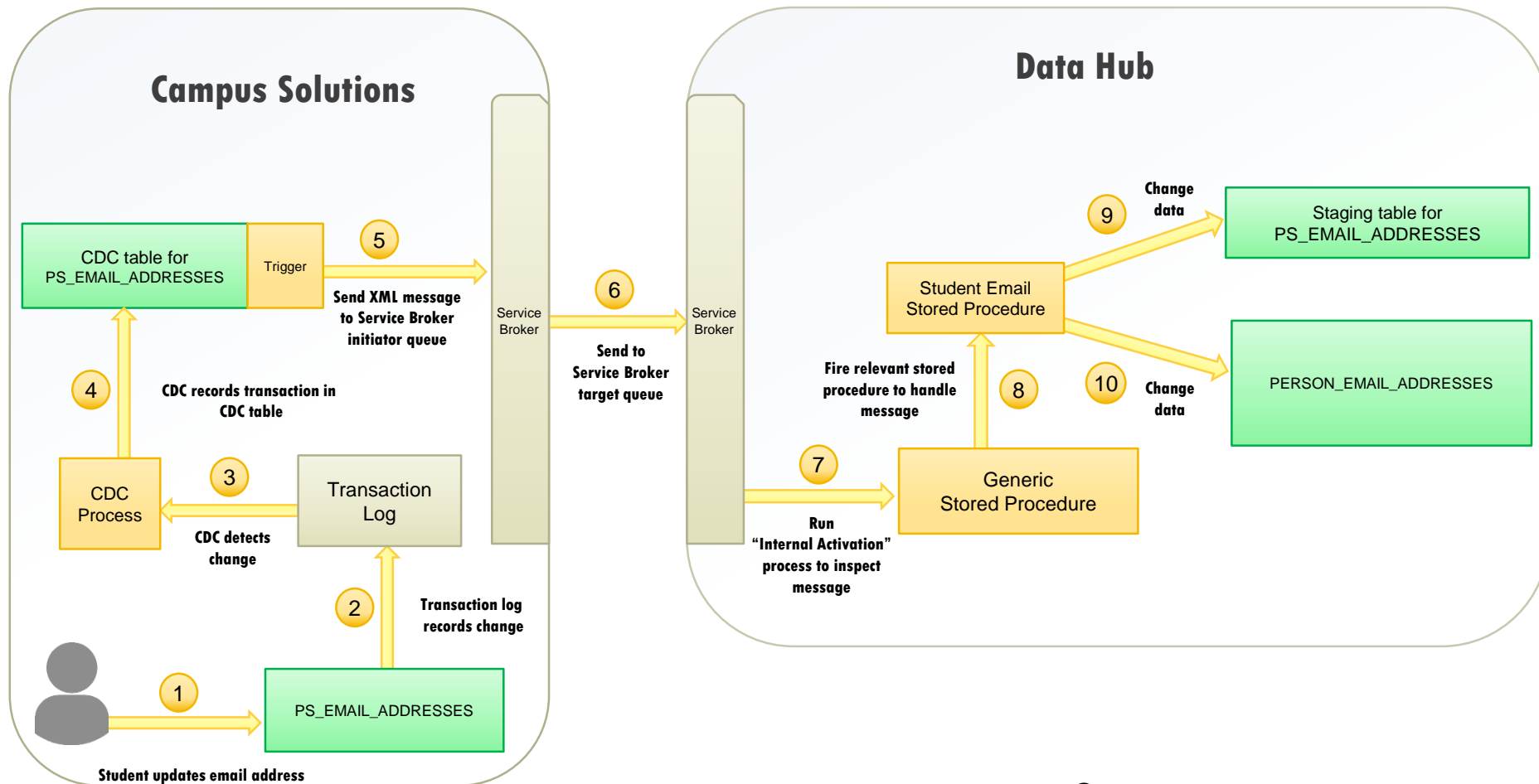
- Leverages Microsoft SQL Server technologies
 - Change Data Capture
 - Provides the ability to track data changes (*add, update, delete*) in database tables
 - PS_ACAD_PROG CDC table add example:

TIME	OPERATION	EMPLID	ACAD_CAREER	STDNT_CAR_NBR	EFFDT	EFFSEQ	ACAD_PROG	PROG_STATUS	PROG_ACTION	PROG_REASON	CAMPUS
2016-11-01 03:00:58.473	Add	1100004	PGRD	1	2016-10-31 00:00:00.000	1	BU710	AP	APPL		SIPPY
2016-11-01 03:00:58.473	Add	1100004	PGRD	1	2016-10-31 00:00:00.000	2	BU710	AD	ADMT	31	SIPPY
2016-11-01 03:00:58.473	Add	1100004	PGRD	1	2016-11-01 00:00:00.000	1	BU710	AC	MATR	ONL	SIPPY
2016-11-01 03:00:58.557	Add	1102440	PGRD	0	2016-10-31 00:00:00.000	1	BU510	AP	APPL		SIPPY
2016-11-01 03:00:58.557	Add	1102440	PGRD	0	2016-10-31 00:00:00.000	2	BU510	AD	ADMT	29	SIPPY
2016-11-01 03:00:58.557	Add	1102440	PGRD	0	2016-11-01 00:00:00.000	1	BU510	AC	MATR	ONL	SIPPY
2016-11-01 03:01:00.310	Add	1074150	UGRD	2	2016-10-28 00:00:00.000	1	AR101	AP	APPL	1PRF	SIPPY
2016-11-01 03:01:00.310	Add	1074150	UGRD	2	2016-10-31 00:00:00.000	1	AR101	AD	ADMT	31	SIPPY
2016-11-01 03:01:00.310	Add	1074150	UGRD	2	2016-11-01 00:00:00.000	1	AR101	AC	MATR	ONL	SIPPY
2016-11-01 03:01:00.393	Add	1098378	UGRD	0	2016-10-31 00:00:00.000	1	SC391	AP	APPL	1PRF	SIPPY

- Service Broker
 - Provides a framework for the reliable transmission of messages between SQL Server databases



REAL-TIME SCENARIO





PERSON INFORMATION

- USC's Data Hub contains data for the following:
 - all students (past and present)
 - active staff
 - active POIs (staff and student)

PERSON IDENTITY DATA



Search for person...



Roger Ramjet (1000134)

Relationships: STAFF, STUDENT, ALUMNI

Data last synchronised at 2/11/2016 1:42:54 PM.

Network Accounts

1	1000134500	Staff Library Account
2	RRAMJET	Staff Network Account
3	r2753494	Student Network Account

Email Addresses

1	r2753494@usc.edu.au	CAMP
2	roger_ramjet@hes_our_man.com	OTHR
3	rramjet@usc.edu.au	STAF

Phone Numbers

1	07 5555 5555	BUSN
2	0400000000	CELL
3	07 9999 9999	PERM
4	07 7777 7777	UNIV

Jobs

Job Number	Position Number	Primary Job?	Job Type	Job Category	Employment Type	Department ID	Manager Position Number	Organisation	Full Time Equivalence	Commencement Date	Termination Date
0	00000007	Y	Staff	APT	Full-time	070	00000533		1.000000	9/03/2015	

Academic Programs

Academic Career	Academic Career Seq	Academic Program	Program Status	Program Action	Program Reason
UGRD	0	RO331 (Bachelor of Rocketeering)	CM	COMP	

Academic Degrees

Degree Number	Degree	Academic Career	Degree Conferral Date
01	RO331 (Bachelor of Rocketeering)	UGRD	20/04/2001

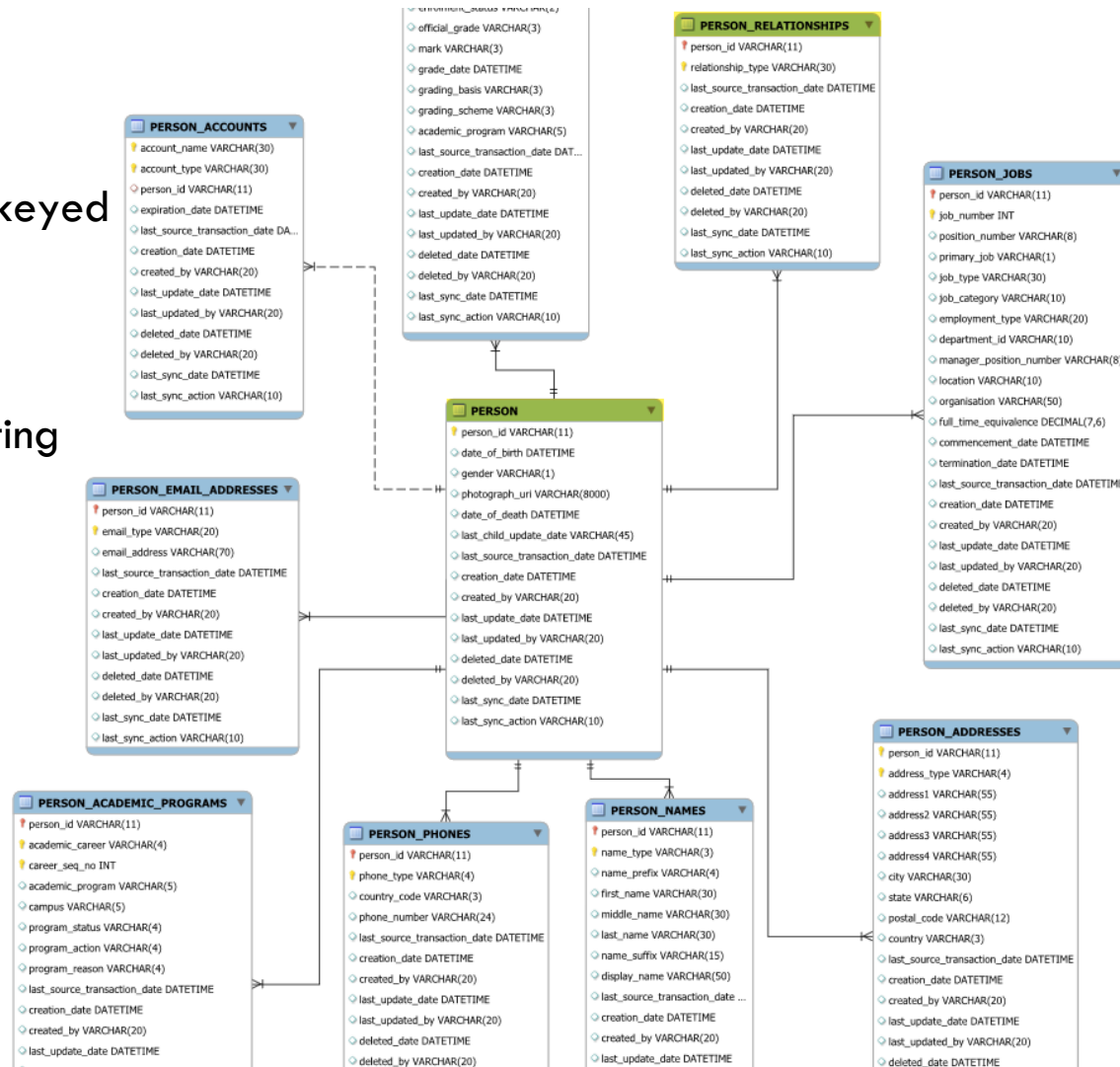


RELATIONSHIP TYPES

Relationship Type	Description
STUDENT	The person is an active student (i.e. currently active or deferred in a program)
STUDENT_POI	The person is a student POI
STAFF	The person is an active staff member
STAFF_OFFEREE	The person has been offered a staff position, but has not yet accepted the offer
STAFF_POI	The person is a staff POI
STUDENT_ALUMNI	The person is a student alumni (i.e. has obtained one or more degrees at the university)
STUDENT_APPLICANT	The person is a student applicant (i.e. is currently applying to study in a program)
STUDENT_OFFEREE	The person has been offered a place in a program, but has not yet accepted the offer
STUDENT_DISCONTINUED	The person was a student that has discontinued an academic program and has no other active programs and has not completed any degrees

DATA HUB SCHEMA

- 36 tables
- PERSON tables keyed by person_id (EMPLID)
- No effective dating





AUDIT HISTORY

Audit Column Name	Audit Column Description
last_source_transaction_date	The datetime that the last transaction was triggered by the source system
creation_date	The date the row was created
created_by	The system that created the row
last_update_date	The date the row was last updated
last_updated_by	The system that updated the row
deleted_date	The date the row was soft deleted
deleted_by	The system that deleted the row
last_sync_date	The date the row was changed in any way (e.g. created, updated, or deleted)
last_sync_action	The last action that occurred for the row (e.g. Create, Update, Delete, Undelete)

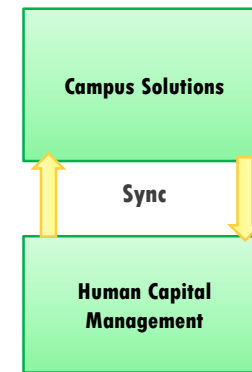


SOFT DELETES

- Rows will never be **immediately** hard deleted (i.e. removed from the table).
- Instead, the `deleted_date` and `deleted_by` fields will be populated for a row when the row has been deleted.
- Rows will eventually be hard deleted after a period of time (currently set to 30 days).

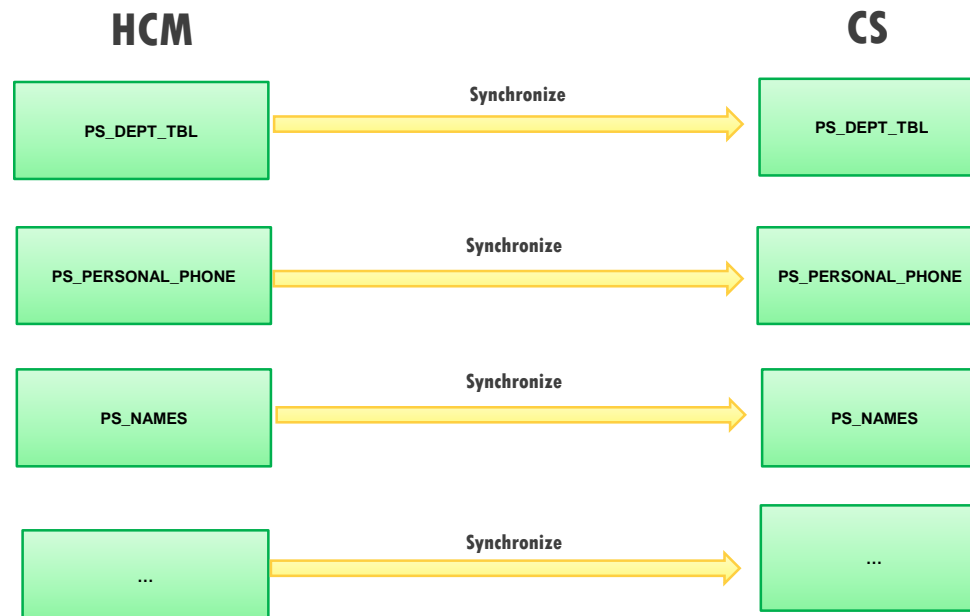
OTHER SYNCHRONISATION PROCESSES

- HCM to CS Sync (real-time and batch)
- CS to HCM Sync (real-time and batch)
- Data Hub Sync (batch run nightly)
 - Handles future dated rows
 - A fall-back mechanism that ensures data integrity whilst protecting against real-time sync threats such as network outages and database downtimes.



HCM TO CS SYNC OVERVIEW

- HCM is the source of truth for staff and department details
- Separate rules for each table to be synchronised. These rules have been approved by the CS and HCM teams.



HCM TO CS SYNC TERMINOLOGY

- **Conservative action** – Only perform an action (e.g. add, update or delete) when the staff member has no other relationship (present or past) with the university (e.g. staff member up until now has never been a student, POI, or student applicant).
- **Non-conservative action** – Perform an action (e.g. add, update or delete) regardless of whether the staff currently has or has previously had a relationship with the university.

HCM TO CS SYNC RULES

Table	Insert Rules	Update Rules	Delete Rules
PS_DEPT_TBL	Insert new rows	Update existing rows	Delete missing rows
PS_NAMES	Non-conservative insert	Non-conservative update	Conservative delete
CITIZENSHIP	Non-conservative insert	Conservative update	Conservative delete
PS_PERSONAL_PHONE	Non-conservative insert	Non-conservative update	Conservative delete
PS_EMAIL_ADDRESSES	Non-conservative insert	Non-conservative update	Non-conservative delete
...



DATA HUB DEPLOYMENT AND MONITORING

An overview



OCTOPUS DEPLOY

- Used to deploy the Data Hub T-SQL code to the HUB and its connected databases for all stages of the development lifecycle (DEV, UAT and PRD).
- Dynamically replaces variables in source code depending on the development lifecycle.
- Uses Powershell to deploy/load the T-SQL into the HUB and its connected databases.

DEPLOYMENT TYPES

A deployment can be triggered in two ways:

- **On an adhoc basis**
 - Primary way to re-deploy the Data Hub code.
 - Developer or DBA logs into Octopus Deploy and triggers a deployment for a certain environment (DEV, UAT, PRD)
- **After a database refresh**
 - HCM and CS refresh scripts issue a call to the Octopus Deploy REST API to directly trigger a Data Hub deployment after a database refresh.

OCTOPUS DEPLOY

[Dashboard](#)[Environments](#)[Projects](#) ▼

Data Hub Deployment



Data Hub Deployment

Create release

Overview

Process

Variables

Releases

Settings

Release

2.0.0-release-
d813f56

Data Hub - DEV

✓ 2.0.0-release-
d813f56
March 8th 2016

Data Hub - SYS

✓ 2.0.0-release-
d813f56
March 8th 2016

Data Hub - UAT

✓ 2.0.0-release-
d813f56
March 9th 2016


Data Hub - PRD


Deploy


Data Hub - CPY

✓ 2.0.0-release-
d813f56
March 8th 2016

OCTOPUS DEPLOY



Data Hub Deployment > Releases > 2.0.1-develop-19fe1f7 >  Deploy to Data Hub - DEV




Data Hub Deployment


[Overview](#)
[Process](#)
[Variables](#)
[Releases](#)
[Settings](#)


[Task summary](#)
[Task log](#)


Task progress


This task started 22 minutes ago and ran for 13 minutes


 Deploy Data Hub Deployment release 2.0.1-develop-19fe1f7 to Data Hub - DEV


 Step 1: Clear Previous Load Report


 Acquire packages


 Step 2: Replace environment variables in source code


 Step 3: Create Deployment Structure


 Step 4: Send Notification Email - Deployment Started


 Step 5: Update Release Information and hub status in HUB database


 Step 6: Load Pre-Deploy Scripts


 Step 7: Load Security Definitions


 Step 8: Load Initialisation Scripts


 Step 9: Load Real-time Integration Sync Library


 Step 10: Load Real-time Integration HR to SIS Sync Code (SIS Code)


 Step 11: Load Real-time Integration HR to SIS Sync Code (HR Code)


 Step 12: Load Real-time Integration SIS to HR Sync Code (HR Code)


 Step 13: Load Real-time Integration SIS to HR Sync Code (SIS Code)


 Step 14: Load Real-time Integration HR to HUB Code (HUB Code)


 Step 15: Load Real-time Integration HR to HUB Code (HR Code)


 Step 16: Load Real-time Integration SIS to HUB Code (HUB Code)


 Step 17: Load Real-time Integration SIS to HUB Code (SIS Code)


 Step 18: Load Real-time Integration SAM to HUB Code (HUB Code)


 Step 19: Load Real-time Integration SAM to HUB Code (SAM Code)


 Step 20: Load Batch Sync Process Code


 Step 21: Load Diagnostics Code


 Step 22: Load Development Tools


 Step 23: Load Interfaces - Databee


 Step 24: Load Interfaces - Corporate Web


 Step 25: Load Interfaces - GCC


 Step 26: Load Post-Deploy Scripts


 Step 27: Perform Health Check

 Step 28: Execute Batch Full Sync

 Step 29: Update hub status in HUB database

 Step 30: Send Notification Email - Deployment Finished

 Step 31: Set Deployment Result

 Apply retention policy on Tentacles



OCTOPUS DEPLOY VARIABLES



Data Hub Deployment

Create release


Overview

Variables

Include variable sets from the Library

Y	Name	Value	Scope
✓	HUB_DATABASE	HUBDEV	DEV
✓	HUB_DATABASE	HUBUAT	UAT
✓	HUB_DATABASE_SERVER	MSJ-SQL-VS99	DEV
✓	HUB_DATABASE_SERVER	MSU-SQL-VS98	UAT
✓	EMAIL_DEPLOYMENT_NOTIFICATIONS_TO	the_boss@usc.edu.au, the_dbas@usc.edu.au, the_developer@usc.edu.au	PRD
✓	EMAIL_DEPLOYMENT_NOTIFICATIONS_TO	the_dbas@usc.edu.au, the_developer@usc.edu.au	DEV, UAT

OCTOPUS DEPLOY — RELEASE NOTES



DashboardEnvironmentsProjects▼LibraryTasksadmin▼Configuration?

Data Hub Deployment > Releases



Data Hub Deployment

Create release

Overview

Process

Variables

Releases

Settings

Releases

2.0.0-release-5e5d782

Assembled Thursday, March 10, 2016 8:26 AM

[2.0.0] - 2016-03-10

Added

REQ0075184 - The Data Hub deployment system now uses Octopus Deploy for both adhoc and release deployments. The old pythonic system is no longer in use. All python code that was used for previous deployments has been re-written in Powershell and is now run from within Octopus Deploy. The deployment server has also been moved from the PeopleSoft database servers to the tfs servers and a new DataHub user has been created to initiate the deployments. The Data Hub T-SQL sourcecode has also been heavily modified to support the Octopus Deploy deployment system.

REQ0075181 - The Data Hub now uses Git for source control. The Data Hub uses the git-flow methodology for Git. Release notes are collated and merged from different stages of development (feature, develop, merge, release) and are eventually merged into the CHANGELOG.md file.

REQ0069262 - The Data Hub now retrieves a person's room location (in real-time) from the SAM system. The room location field is stored in the sam_location_code of the PERSON_ROOM_LOCATIONS table. The PERSON_ROOM_LOCATIONS key structure allows for multiple sam_location_code values for a person. Service Broker and Change Data Capture have been enabled on the SAM databases. The existing PERSON_LOCATIONS table has also been modified to store the campus and location values for a particular relationship_type.

REQ0075187 - The academic_organisation and split_ownership fields have been added to the ACADEMIC_PROGRAMS table in order to provide the ability to identify single ownership academic organisations for a particular academic program.

REQ0075780 - A new HOLIDAY_SCHEDULE_DATES table has been added to the Data Hub to store date ranges for particular holiday schedules. Also, the CLASS_MEETING_PATTERNS table now contains additional metric fields in order to assist in providing more detailed reporting of class schedules.

REQ0071895 - New SIS to HR synchronisation processes (in both Real-time and Batch) have been created. Like its HR to SIS counterpart, the Batch process is initiated from the Data Hub Batch Full Sync process. At this stage, only the PS_LOCATION_TBL table is synchronised from SIS to HR.

Changed

REQ0071896 - The Batch HR to SIS Department delete logic has been modified such that it will only delete effective dated rows that are missing for the departments that are shared between the two systems. The Data Hub will not delete any *exclusive* department rows in SIS that don't exist in HR. The Real-time HR to SIS Department delete logic will not change, since real-time deletes are in most cases intentional and thus should continue to flow through from HR to SIS.

REQ0075186 - The Data Hub AD group and user security model has been redesigned. All security definitions for hub objects (stored procedures and hub tables) are now stored in the HUB_SECURITY_TBL table. During a deployment, access is granted to the AD groups and users at the time of creation of the object rather than at a later deployment step. This security model allows for deployments to be run at any time without affecting the security permissions of any subscribing systems. It also allows new users to be granted permissions immediately without requiring any further deployments.


REQ0075183 - All tables that contain the term field as part of their key, will now only retain data from the start of the previous year.




HEALTH CHECK PROCESS

- A stored procedure that is run:
 - Nightly
 - During a deployment
- Verifies integrity of Data Hub and all connected systems by:
 - Checking for any SQL errors reported by the Data Hub
 - Checking for missing triggers on connected systems
 - Sending test messages from each connected system
 - Running unit tests (on development databases only)

HEALTH CHECK - EMAIL NOTIFICATIONS

 datahub-deployment@usc.edu.au

DG DBAdmins; Jay Mathew; 

26/07/2016

HUBPRD - Data Hub real-time monitor alert

 Click here to download pictures. To help protect your privacy, Outlook prevented automatic download of some pictures in this message. HUB_HEALTH_CHECK...
482 bytes HUB_ERROR_TBL.csv
501 bytes

Action Items

+ Get more add-ins

Data Hub monitor alert

The following shows the Data Hub monitor status for database **HUBPRD**.

Health Check status: 

Errors have been recorded in HUB_ERROR_TBL table.

Please see the attached file(s) for specific errors to help diagnose the issue further.



CHALLENGES AND NEXT STEPS

Where to from here?

CHALLENGES

- Service Broker learning curve
 - Examples provided by Microsoft are very simplistic and impractical
 - Heavy reliance on SQL for Service Broker administration
 - Lacking UI utilities for debugging
- Slow service broker queues during open enrolment periods



NEXT STEPS

- Enterprise Service Bus Integration
 - Publishing notifications from the Data Hub to the cloud
 - Web service support

PRESENTERS

Jay Mathew

Senior Business Systems Analyst

University of the Sunshine Coast

jmathew@usc.edu.au

**ALL ALLIANCE PRESENTATIONS WILL BE AVAILABLE FOR
DOWNLOAD FROM THE CONFERENCE SITE**



THANK YOU!



ADU 9-11 NOVEMBER 2016