




2020 February Summit

Summit @ San Antonio Riverwalk
February 12-14, 2020

Please send feedback and future discussions to:
melissa@digonsystems.com



1



CMMS Integration

Closing the Loop

Presented by Melissa Nagel



2



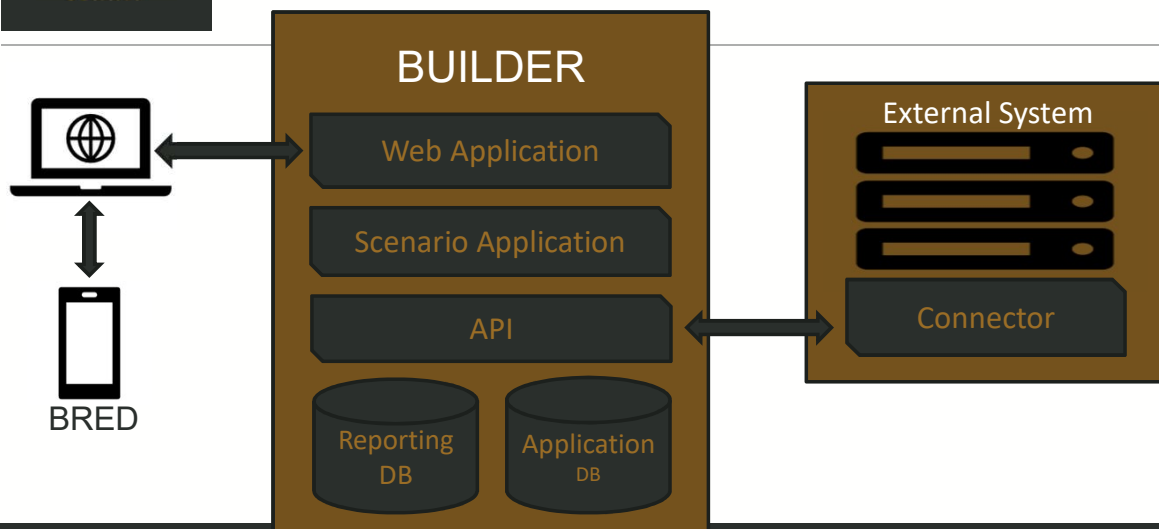
The Challenges

- Many systems managing different aspects of the same or related data
 - Facility data
 - Inventory/equipment data (master equipment lists)
 - Condition assessment data
 - Work order/preventive maintenance management
- Keeping data in sync
- Manual data entry is time-consuming and error-prone
- Security

3



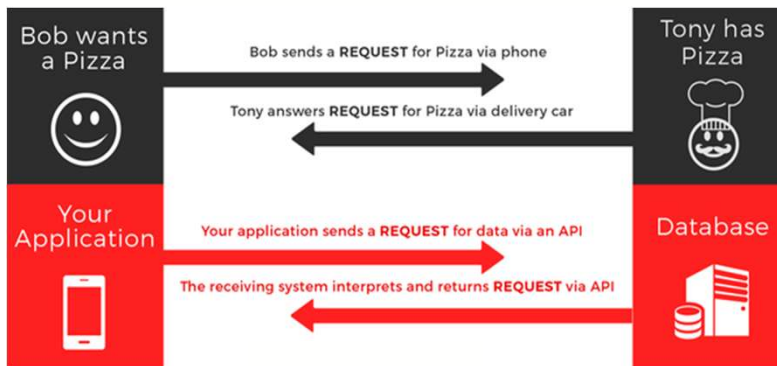
BUILDER System View



4



API Introduction



- Application Programming Interface
- System that enables communications between computer systems

5



Current BUILDER API Capabilities

- Buildings
- Inventory (Systems, Components, Sections)
- Inspections
- Work Items
- Scenarios (limited)
- And more

6



Case Study - NNSA

- NNSA partnered with DIGON to integrate their systems with BUILDER
 - Leveraged our experience exchanging data from our field assessment tool to BUILDER via BRED syncs
 - Developed SPIRE, a platform that sits between NNSA's systems and the BUILDER API
- SPIRE has been used to facilitate integrations of a range of systems, including:
 - Maximo, Asset Suite, Infor EAM, and others at NNSA
 - RSMears
 - iEMS, proprietary system developed by Air National Guard

© 2018 DIGON Systems. All Rights Reserved.

7

7



Why Not Go Directly Through the BUILDER API?

- 9 sites, many different systems—SPIRE offers a common data structure and set of messages that enables multiple systems to integrate in parallel
- CERL is moving quickly, focused on releases
 - Can be challenging to keep up with API changes and enhancements
 - As releases happen, we can develop & test against them so it doesn't break NNSA's existing integration

© 2018 DIGON Systems. All Rights Reserved.

8

8



Why Not Go Directly Through the BUILDER API?

- Simplifies integration logic
 - BUILDER API is focused on individual requests; SPIRE puts it all together into a single request
 - Ordering pizza...make one call
 - To get data on all Buildings in your Site, must iterate through all individual calls:
 - ➔ Get all orgs
 - ➔ For all orgs, get all sites
 - ➔ For all sites, get all complexes
 - ➔ For all complexes, get all buildings...

© 2018 DIGON Systems. All Rights Reserved.

9

9



Why Not Go Directly Through the BUILDER API?

- Single point to implement business rules across organization
 - NNSA buildings (FIMS) example
- High-security environment
 - SPIRE adds logging and a “low-tech” option to enter batch updates using CSV files (Web Portal)

© 2018 DIGON Systems. All Rights Reserved.

10

10



Power of the API

- Live demo – Creating 200+ buildings

11



Laying the Foundation

Decide how your CMMS will communicate with another API (BUILDER, SPIRE, etc.)

- CMMS may have a scheduled data export function that you'll need to figure out how to relay on to BUILDER/SPIRE API
- CMMS may have API integration functionality built in
- CMMS may have its own API endpoints that you need to create a middle-tier service to connect to BUILDER/SPIRE API
- CMMS may not have any integration capabilities and you need to perform custom development to create a solution
- Consult your CMMS provider or integration partner

12



Laying the Foundation

Map fields between BUILDER and other systems

- Build a data ownership map
 - System of record - Decide which system is the source of truth for each field or type of data (e.g., FIMS for facility data)
 - Path of data flow
- Identify which fields you want to include in the data exchange
- Identify keys to link other systems to BUILDER



Laying the Foundation

SPIRE Name	BUILDER Display Name	Maximo	InSite/Infor EAM	EquipHub	SAP - CIC
EquipmentId	ID Number	<Asset Number>	<Unique ID> (MEL / CAS) Non-MEL will have another unique ID	<Revit Element ID>-<Model>-<Building No>	For new records, populate with the SAP <EquipID>
BlghId	Alternate ID				New mapping table in CIC to align RPUID
SectionCatalogId	<not displayed in BUILDER>			Available in the EquipHub logic	The Mapping Decision Table to allow Devs to select the Catalog drop down.
SectionName	Section Name	<Floor>-<Room>-<Quad>-<Asset Number>	<Building>-<Floor>-<Room>-<Unique ID>	Need to refine based on section year	<Equip ID> Auto populated with free text added (Room number, Floor, etc.)
SectionQuantity	Quantity	<Units>	<Quantity>	Converted in the logic based on the SMS UOM	<User Input>
SectionYearBuilt	Year Install/Renewed Year Built/Renewed	<Install Date>	<Install Date>		<Date Created>
SectionEstimated	Estimated				
SectionRSL	RSL				
RemainingPaintLife	RPL		Not Used		
SectionsPainted	Painted		Not Used		
SectionDatePainted	Year Painted		Not Used		
SectionComments	Comment		Possible holding point for notes on details		Will be using BUILDER for comments
SectionCI	CSCI				
SectionCCI	CCI		Not Used		
EquipmentType	Eq. Type		<Equipment Type>		<Technical Object Type>
EquipmentMake	Eq. Make		<Equipment Make>		
EquipmentSerialNumber	Serial No.	<Serial Number>	<Serial Number>		<Serial No>
EquipmentModel	Model	<Model>	<Control_Type_Make>		<Model>
EquipmentCapacity	Capacity		Concatenate list		
EquipmentManufacturer	Manufacturer	<Manufacturer>	<Manufacturer>		<Manufacturer>
EquipmentWarrantyCompany	Warranty Company	<Warranty Company>	<Warranty Company>		<Manufacturer Location>
EquipmentWarrantyDate	Warranty Date	<Warranty Expiration Date>	<Warranty Date>		
EquipmentWarrantyCompany2	Warranty Company 2		Not Used		
EquipmentWarrantyDate2	Warranty Date 2		Not Used		
EquipmentLocation	Location	<Room>	<Room Number>-<Location>	<room number>	<Location>
EquipmentManufactureDate	Date Manufactured		Not Used		
EquipmentControlType/Make	Control Type/Make				
EquipmentComments	Comments		<Comments>		<Description>
EquipmentCountMultilinear	Count displayed in RIII DFAs				



Laying the Foundation

Translate data fields from BUILDER to other systems

- Some Options
 - Store BUILDER data natively in your system (import the BUILDER catalog into your CMMS--method depends on CMMS)
 - Create a mapping table that allows you to map BUILDER values to your CMMS system's native values, and you perform the translation on import/export



Key Lessons Learned

NNSA

Sandia and Nevada (Maximo)

- Using built-in Maximo integration tools; didn't need to create a complicated middle-ware solution
- Cost-effective approach

LLNL (Infor EAM/InSite)

- InSite→FIMS/MEL/CAIS-CAS/Infor EAM/BUILDER
 - In transition between CMMSs (SAP)
 - Built an in-house app/database to facilitate communication between all these systems
- Chose a flattened inventory approach
 - 1 section detail per section



Key Lessons Learned

NNSA

LANL (Asset Suite/NextAxiom)

- Data structure different in Asset Suite than BUILDER
 - Aligned their Asset Suite Master Equipment List with Uniformat
 - Challenge: MEL broke everything down into individual components with no parent/child relationship.
 - Example:
 - HVAC 5 -ton (ecode)
 - Belt (ecode)
 - Motor (ecode)
 - ... could have 20+ ecodes for a single BUILDER section
 - Section Alt Id good alternative for grouping these in BUILDER
- NextAxiom
 - Example of middleware option to communicate between CMMS and SPIRE/BUILDER API
 - Also provides a “midpoint” review area before changes are pushed

17



Key Lessons Learned

NNSA

KCP (EquipHub)

- BIM-loaded inventory/equipment
- Results in multiple section details, even for non-EA sections
- Required SPIRE enhancements to support complex multiple section detail logic
 - E.g., changing section-level info, adjusting quantities, creating/updating inspections, etc.

18



Key Lessons Learned

Gordian

- RS Means—cost data
 - Pull inventory and modifiers
 - Calculate PRVs
 - Push updated PRVs
- Non-CMMS system
- Example of partnership to create a custom message to meet their specific needs; completely proprietary system on Gordian's end



Key Lessons Learned

ANG/iEMS

- Work order integration (Work Request ID)
- Populating EquipmentIds through assessments
- Phased approach
 - Automated data exports to get the right data structures
 - Manually upload through SPIRE CSVs to validate data structures
 - Phase-in full system integration



Wrap-Up

Invest time laying the groundwork

- Involve Information Assurance / Cyber Security early
- Maintain open lines of communication between all stakeholders, especially existing CMMS users
 - In-person meetings are invaluable—at beginning and then periodically throughout project
- Benefits
 - Fewer surprises and smoother implementation phase
 - Framework for decisions as they arise during development
 - Avoid unnecessary rework



Resources

- Contact CERL for documentation targeted to a software developer audience
- Identify support resources for your non-BUILDER systems (Maximo, Asset Suite, etc.)
- Partner smartly
 - BUILDER can be quirky so leverage those with an understanding of the system
 - Look for BUILDER expertise and technical know-how that can help bridge the gap between your existing systems and BUILDER