GenRocket

GENROCKET ENTERPRISE DEPLOYMENTS

PRE-DEPLOYMENT PHASE

1. **Assessments**

   Application Environment

   Understand the applications and infrastructure that supports them and underlying databases. Deliverable = Applications Audit: Identifies applications and infrastructure

   Data Environment

   Understand the information flow through the application environment and the data dependencies. Deliverable = Data Audit: Identifies data flow and data dependencies

   Testing Environment

   The current testing resources, test processes, testing tools and data provisioning processes. Deliverable = Test Audit: Test data tool and framework integrations.

   Enterprise Deployment Plan

   The TDG deployment plan for the organization.

   Deliverable = Enterprise Deployment Plan: Prioritized plan for application testing, new processes for test data provisioning, data refresh and test data integration with tools and frameworks.

2. **Identify resources to match Roles & Responsibilities of Project**

   Roles & Responsibilities

   Project Manager

   Oversees and manages the TDG implementation project.

   Data Architect

   Understands the data model and relationships between the different applications and databases.

Solution Architect

Understands the testing organizations test data needs and as a GenRocket Certified Engineer understands how to set up and configure GenRocket to meet the test data requirements.

Data Specialist / Testers

Uses pre-built test data generation Scenarios to generate test data for different test cases.

Builds and modified test data generation Scenarios to meet test data needs for different test cases.

3. Training

GenRocket University

GenRocket Certified Engineer Course – Data Architects & Solution Architects.

- Invested time 10-20 hours.
- On-line training with 2 hands-on assignments.

Power User – Data Specialists

- Invested time 3-6 hours.
- On-line training with 1 hands-on assignments.

Standard User - Testers

- Invested time 2-4 hours.
- On-line training with 1 hands-on assignments.

4. GenRocket System Administration Prep

Administration Steps

Identify GenRocket Admin

- Admin can add and remove users within the organization.
- Admin can choose level of system access on a per user basis.
- Admin has ability to see all system reports within the organization.

Identify GenRocket Users

- Data specialists
- Testers
- Etc.
- Will need User's First Name, Last Name and organization email address.

Select level of system access for each user

- Admin level – organization wide system access
- Project level – Only able to access specific Projects
- Domain level – Only able to access specific Domains
- Scenario level – Only able to access specific Scenarios

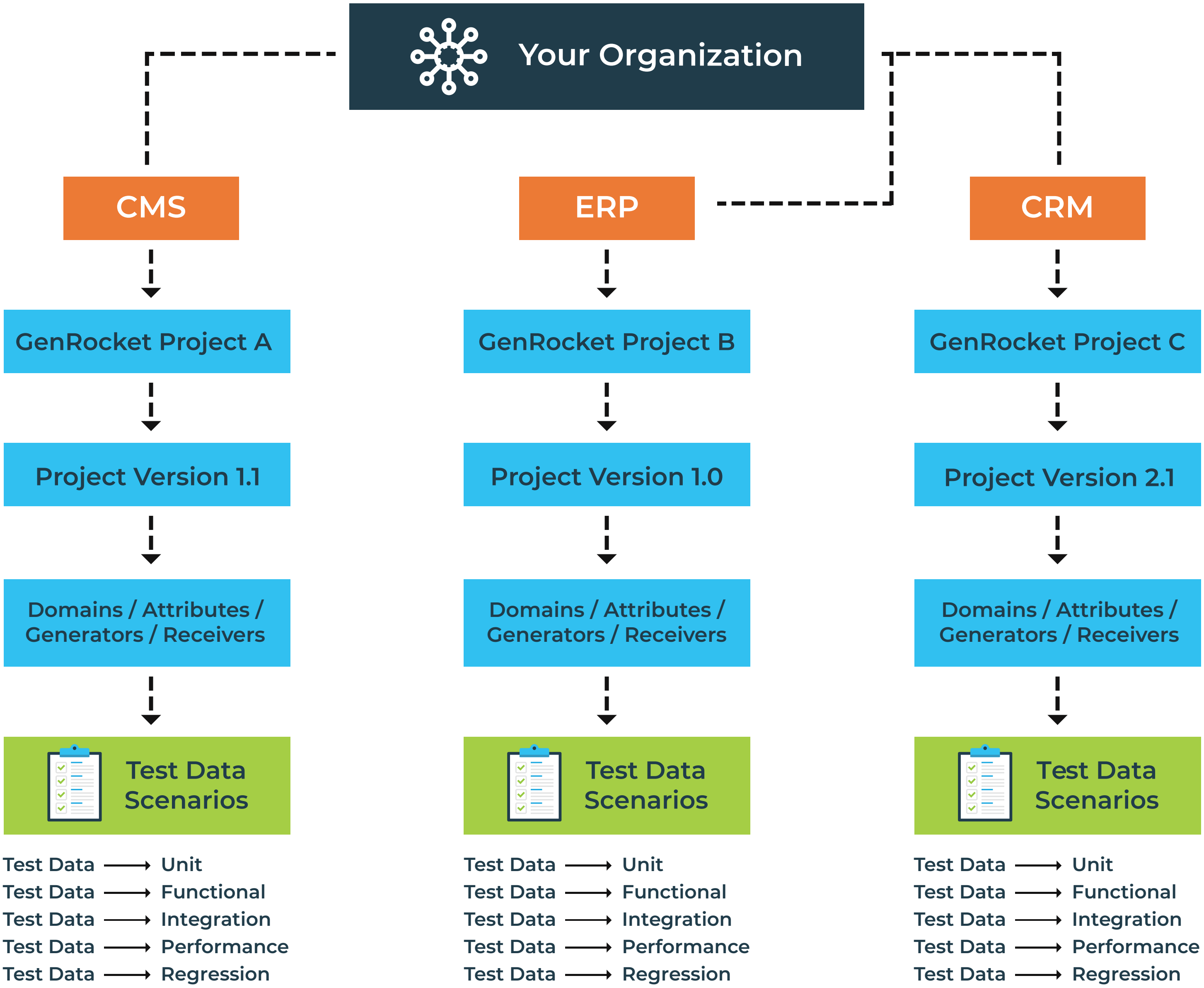
DEPLOYMENT PHASE

1. **Enterprise Deployment Plan**

   Use the prioritized deployment plan that was created in the pre-deployment phase and start with the most important applications first.

2. **Deployment Approach**

   Deployment is usually one GenRocket Project per application or database..



3. **Use XTS (Extract Table Schema) for fast setup of the application or database in the Project.**

   Data Architect or Solutions Architect will use XTS.

   - Import the Data Schema from the application / database.
   - Use GenRocket Wizard to establish Parent-Child-Sibling relationships.
   - In the background the GenRocket data warehouse will automatically scan each Attribute and will auto assign Generators to every Attribute
   - Generators that were incorrectly assigned to an Attribute will need to be swapped for the correct Generators and some of the Generator parameters will need to be modified (e.g. First Name changed to Last Name)
   - Assign Receivers to the Domains (in groups) so that data is generated in desired output format(s) (e.g. XML, JSON, SQL)

   Solution Architect or Data Specialist or Testers

   - Build test data generation Scenarios to meet test case requirements.

4. **Think Differently About Test Data**

   No more Gold Copy, No more Data Refresh.

   - Traditional test data thinking is that Gold copies of the entire database need to be created and then the data needs to be subset into usable test data sets for test cases. Then the data gets stale and needs to be refreshed. This is not the approach to take with GenRocket.
   - Because GenRocket Scenarios generates data at a rate of 10,000 rows per second or can generate billions of rows in minutes using the Partition Engine there is no need to store, manage or refresh test data in databases, files, or spreadsheets. Data never needs to go "stale" when it is generated so fast. Generate the data when it is needed, on demand, and then discard the data when the test has completed.

   Self Service vs. Request and Wait.

   - Instead of a "test data request and wait" model you can move to a "self service" model where test data specialists and testers create and modify their own test data generation Scenarios and Scenario Chains to provide test data for their test cases. Each tester gets their own GenRocket license and through a Web browser accesses GenRocket Web to build test data generation Scenarios for each test case.
   - GenRocket also has the ability to support a self-service model where data is requested from within a testing automation framework and data is dynamically provisioned in real time. In this model the tester doesn't actually know they are making a request to GenRocket they just know they make a test data request and the system returns the data they need for their test. This version of self service involves customization and integration using the GenRocket Multi User Server (GMUS) and the GenRocket API.

   Tag, Categorize and Share Scenarios

   - GenRocket Scenarios can be tagged and categorized so that commonly used test data Scenarios can be created and easily shared between testers. Scenarios are currently stored in a GenRocket folder on each tester's computer or in a folder on the testing automation server but when GenRocket Repository (GRepository) is released (April 2019) it will be a central repository for all GenRocket Scenarios. This new model for storing Scenarios will make it easier to update all Scenarios when there are data model changes within the Project.

Dynamic Data Generation: The GenRocket API

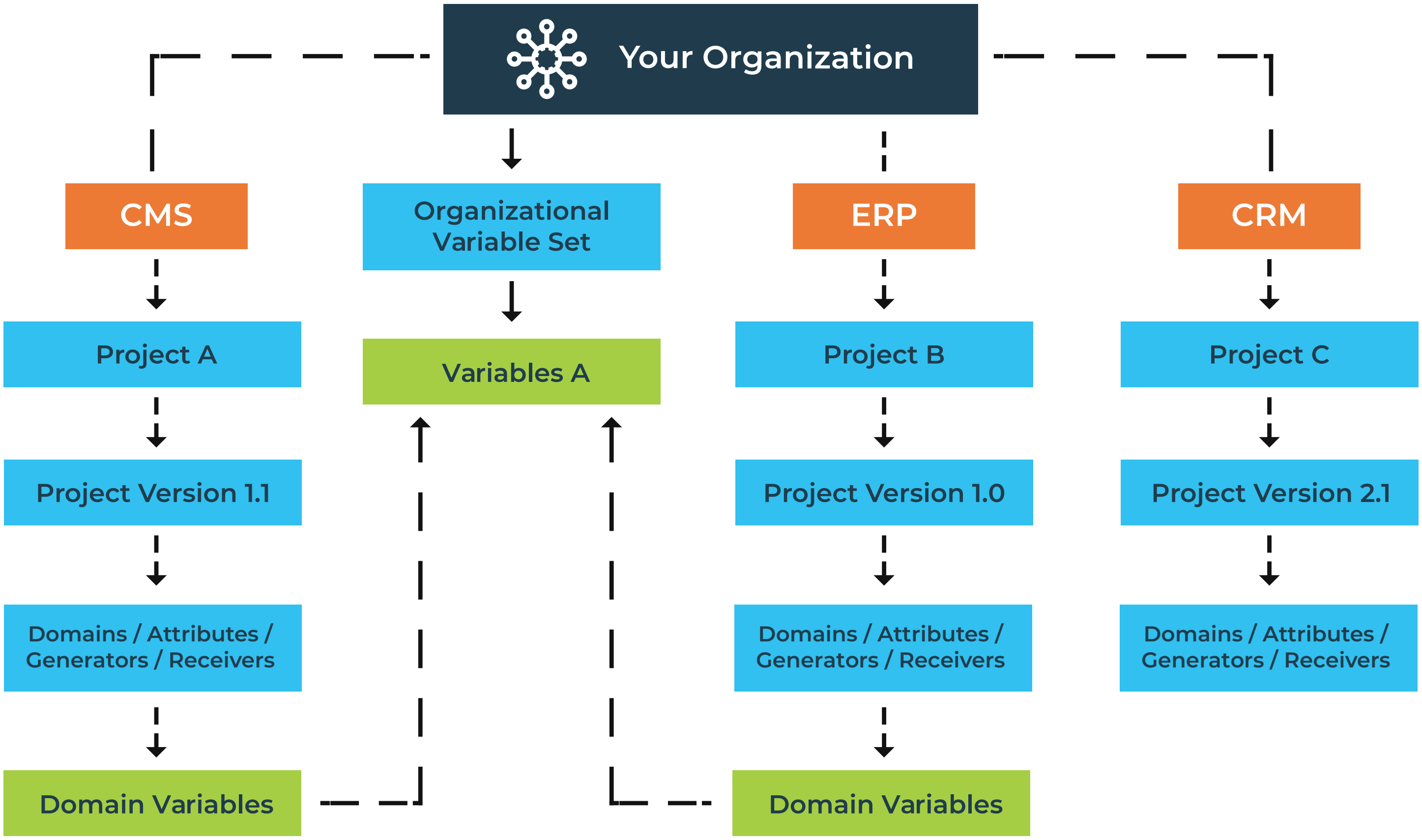
- In addition the GenRocket API can be used to dynamically generate data based on rules and conditions. The API can start with a blank Scenario (no Domains, Attributes, Generators or Receivers) and then instantly add them based on the requirements of the test case. This is a more advanced use case but allows for very powerful and complex testing scenarios.

Referential Integrity

- When referential integrity is needed for a test case, for example. Organization, Department, User and Address (four Domains) there will be one data generation Scenario for each Domain and the Scenarios will be grouped into a “Scenario Chain” so that all the data is generated together for the test case.

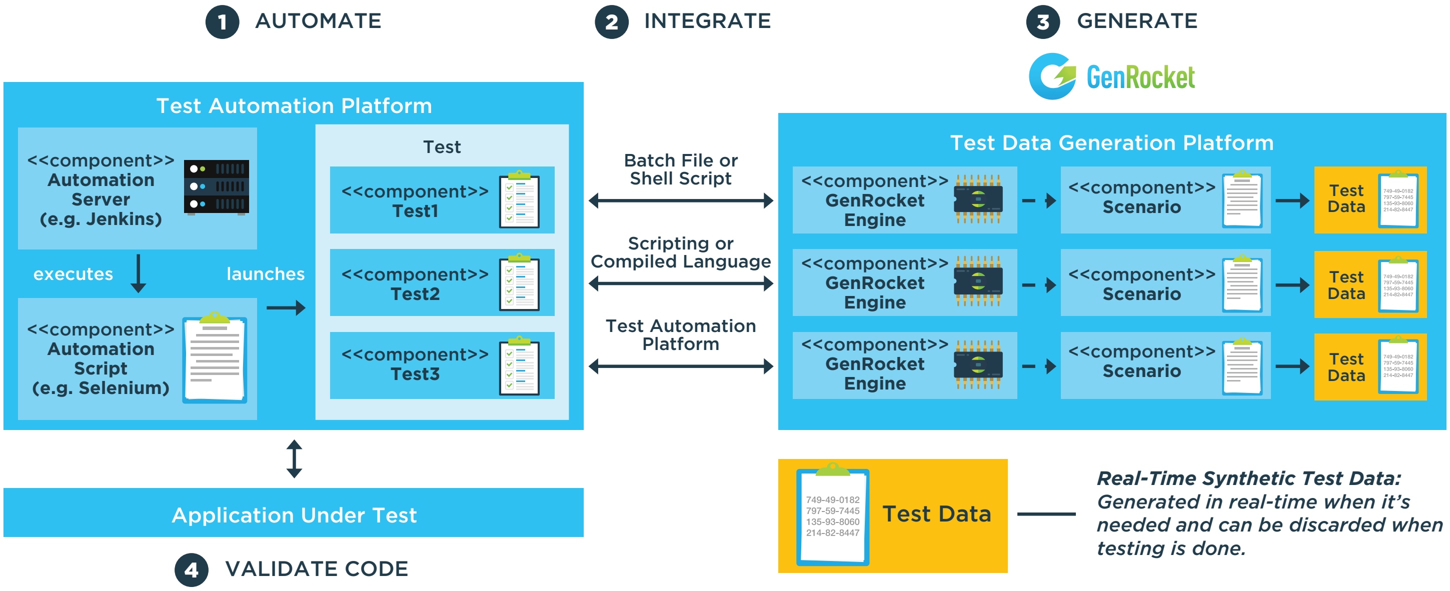
5. Establishing Referential Integrity between Projects

GenRocket has several approaches to ensure that there is referential integrity between Projects. One approach is to use an Organization Variable Set as shown below


Your Organization
CMS
Organizational Variable Set
ERP
CRM
Project A
Variables A
Project B
Project C
Project Version 1.1
Project Version 1.0
Project Version 2.1
Domains / Attributes / Generators / Receivers
Domains / Attributes / Generators / Receivers
Domains / Attributes / Generators / Receivers
Domain Variables
Domain Variables

6. Integration with Testing Automation Tools

While most testing automation tools offer the ability to integrate test data from a CSV file there are usually several other ways to integrate GenRocket test data into the different tools. See the diagram below for examples...



1 AUTOMATE
2 INTEGRATE
3 GENERATE
GenRocket
Test Automation Platform
<<component>> Automation Server (e.g. Jenkins)
executes
launches
<<component>> Automation Script (e.g. Selenium)
Test
<<component>> Test1
<<component>> Test2
<<component>> Test3
Batch File or Shell Script
Scripting or Compiled Language
Test Automation Platform
Test Data Generation Platform
<<component>> GenRocket Engine
<<component>> Scenario
Test Data
<<component>> GenRocket Engine
<<component>> Scenario
Test Data
<<component>> GenRocket Engine
<<component>> Scenario
Test Data
Application Under Test
4 VALIDATE CODE
749-49-0182
797-59-7445
135-93-8060
214-82-8447
Test Data
Real-Time Synthetic Test Data: Generated in real-time when it's needed and can be discarded when testing is done.

GenRocket also has several Realtime Engines that can be used when generating test data from programming languages or for testing automation tools that are not based on the Java Runtime. Both of the following engines are language agnostic.

- **Realtime Socket Engine:** read the knowledge base article for more info
- **Realtime REST Engine:** read the knowledge base article for more info

Need more information about our enterprise deployments? Let's chat!
Send an email to **info@genrocket.com** to start the discussion.

About GenRocket

GenRocket is an emerging technology leader in software testing technology, serving IT services companies and enterprise customers who demand superior quality and efficiency in their software development operation. GenRocket is the next generation of Test Data Management (TDM) solutions that uses a synthetic test data generation system that allows users to generate test data on-demand and in real-time. This system was built for enterprise use cases and can scale to large, complex database environments while preserving referential integrity.

GenRocket, Inc.
2930 East Ojai Ave.
Ojai, CA 93023, USA

1-805-836-2879

info@genrocket.com