



DataView Suite

The Face to your Test Data



PROBLEM STATEMENT

Ensuring we have provisioned and can find “Fit for Purpose” test data is difficult. Without this ability we are exposed to low test productivity & significant rework!

CHALLENGES

Understanding & Mining for test data is complex

- Rarely is data in one place (many apps, many DBs)
- Architectures and Data Sources are typically diverse

Cross platform searches are time consuming

- Fulfilment often dependent on many Application/DB SMEs
- Mining can be difficult, hard to synchronize or repeat

Managing the organisations test data is non-Trivial

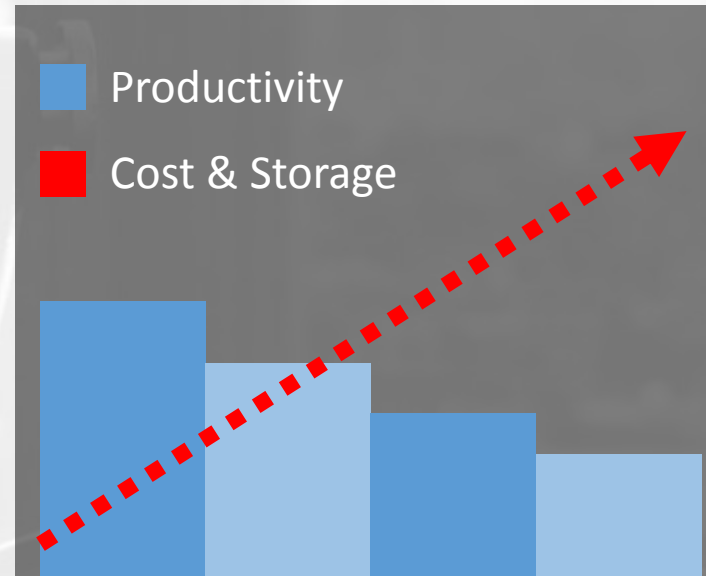
- Difficult to track/identify data requirements or data usage
- Difficult to understand test case and test data relationships

Too few test environments cause data contention

- Contention increases corruption & Data Related Defects
- Contention causes an agile fall (execution being blocked)

Data Issues impact overall testing productivity

- Less Productivity means weaker exercising of solutions
- Lack of agility results in costly delays to your projects



Industry surveys have identified that, on average, **20%** of Test Cycle productivity is lost due to Test Environment & Test Data challenges.

DATAVIEW IS THE “FACE TO YOUR DATA”

The DataView Suite promotes DevTester agility through the use of Test Data Accessibility & Test Data Governance Mechanisms.

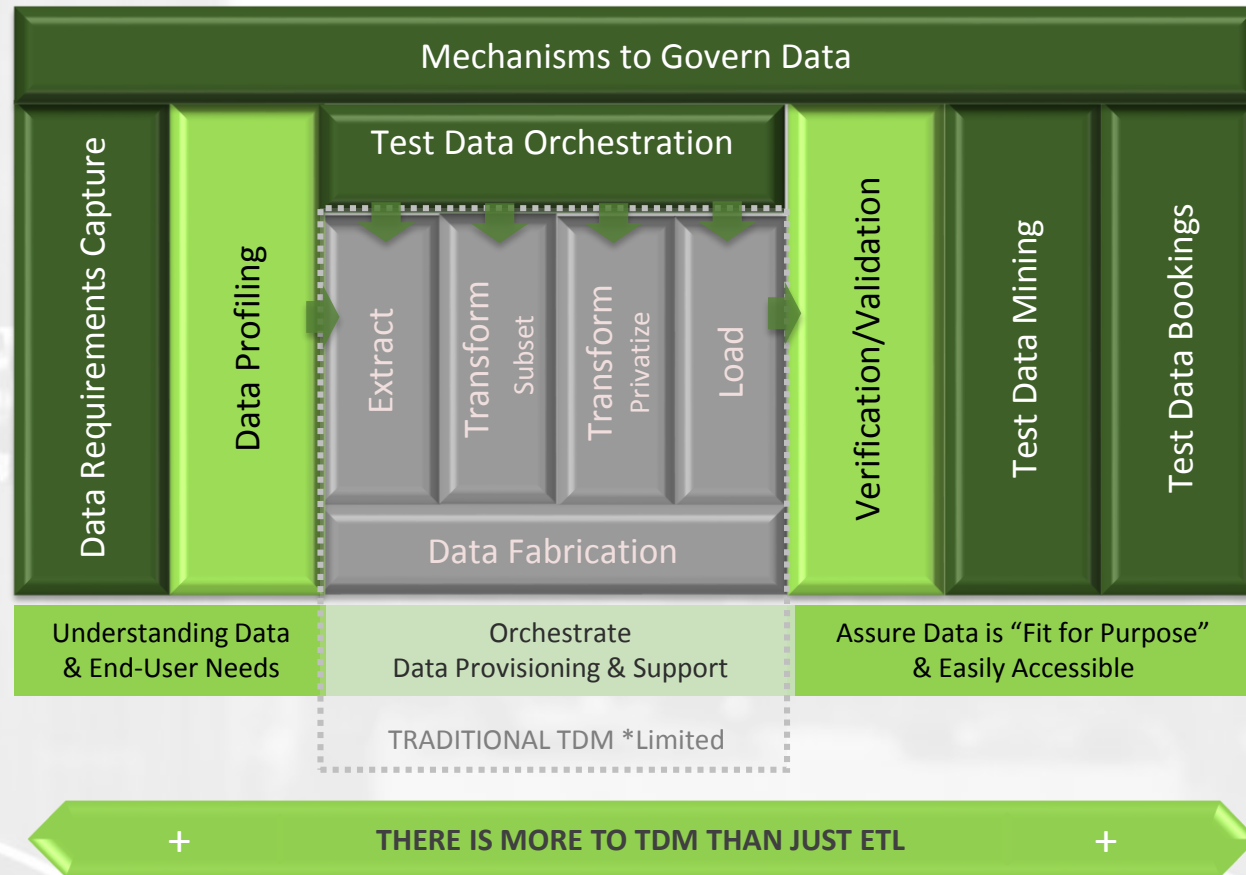
Key Features of DataView Suite include:

- Test Data Requirements Capture
- Test Data Profiling
- Central Data Orchestration
- Test Data Validation *Security Checks
- Rapid Test Data Mining
- Test Data Bookings
- Data Usage Analytics



DataView does not replace your existing provisioning mechanisms!
DataView was designed to uplift your existing Test Data Management methods and promote a more holistic and mature capability.

A Holistic Test Data Management Framework



OPERATIONAL OVERVIEW

The DataView approach can be described in 4 simple steps:

1. Point DataView at your key Test Data Sources

- Applications may be hosted on Windows, Unix, Mainframe etc.
- E2E Data sources may include Oracle, MSSQL, DB2, IMS, Files etc.

2. Design & Build Data Views that DevTesters understand

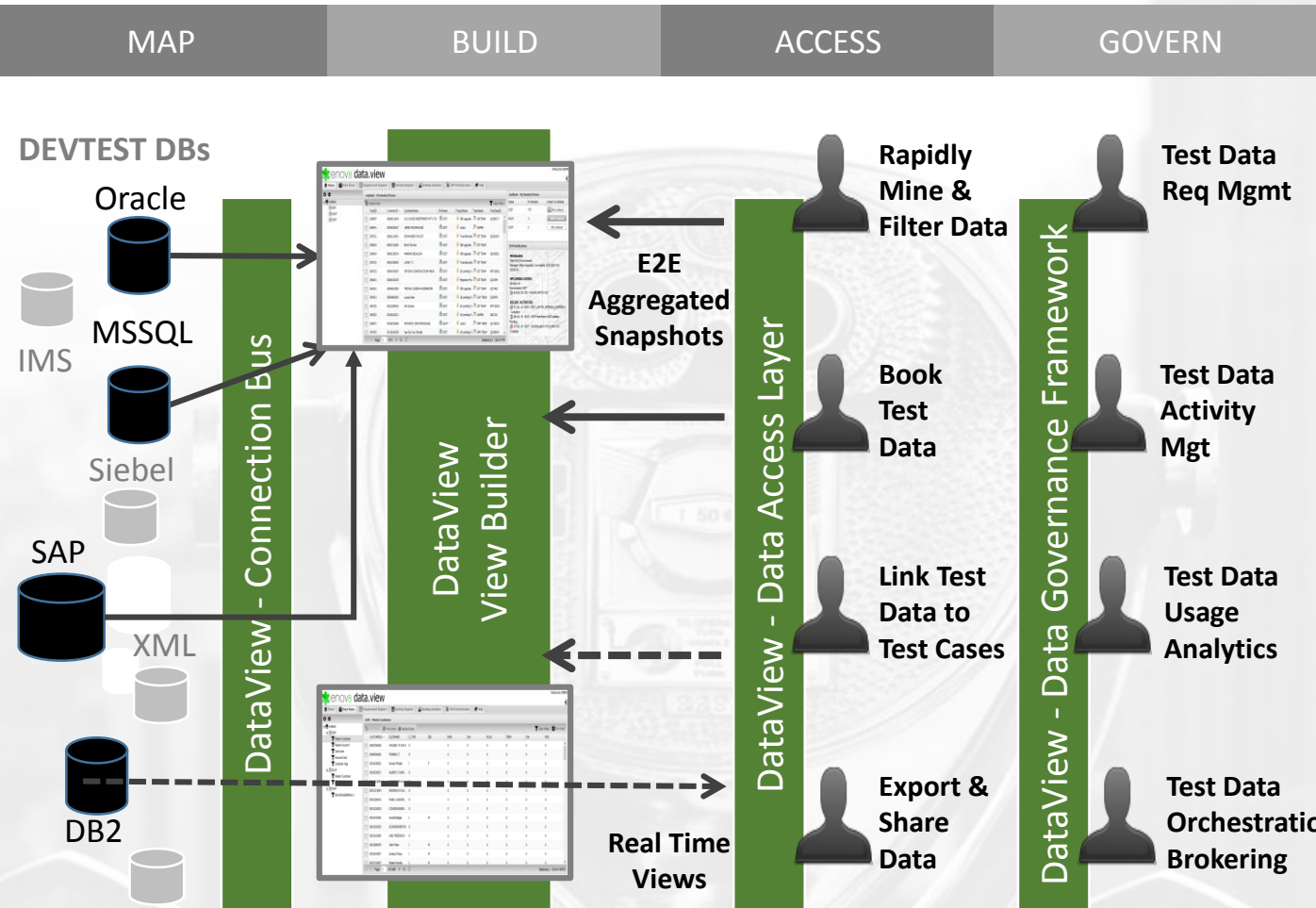
- Select E2E data that will make your DevTesters most productive
- Create “aggregated” views that show E2E data and relationships

3. Have DevTesters proactively “self service” that test data

- Have DevTesters filter and/or mine their own test data
- Have DevTesters book/reserve data for “safe” testing

4. Use embedded mechanisms to better manage the data

- Features to capture Test Data Requirements
- Mechanisms to link Test Data with Test Cases/Assets
- Mechanisms collaborate Test Data activity
- Means to hide Non Fit for Purpose Data
- Automation methods to orchestrate Test Data Tasks (e.g. ETL)



KEY BENEFITS

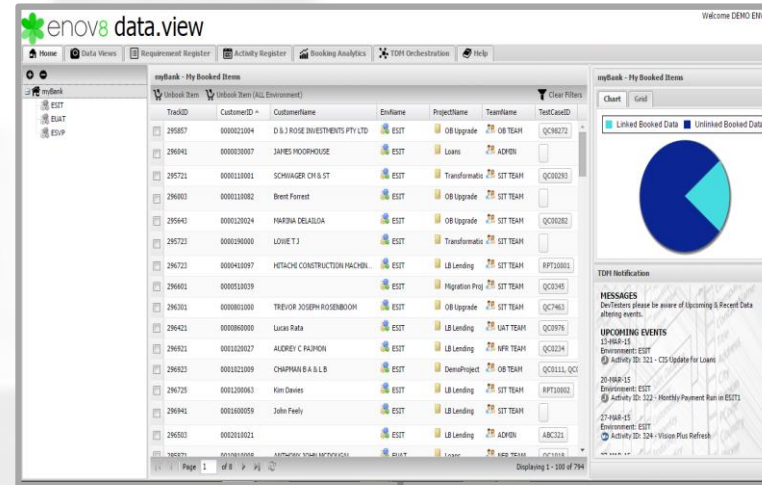
- Testing productivity through rapid accessibility
- Testing productivity via rapid “validation” of data
- Tester up-skilling via data (& relationship) simplification
- Safe sharing of data (contention avoidance)
- Proliferation avoidance through data usage analytics
- Improved Securitization via profiling & validation methods

SUPPORTED DATA SOURCES

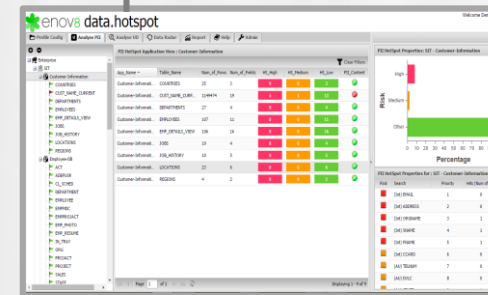
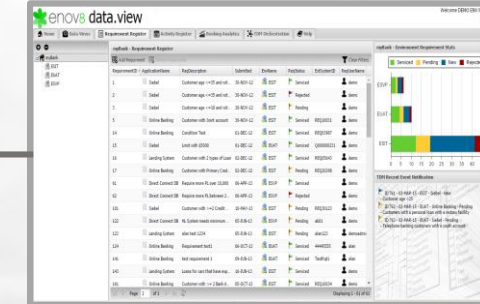
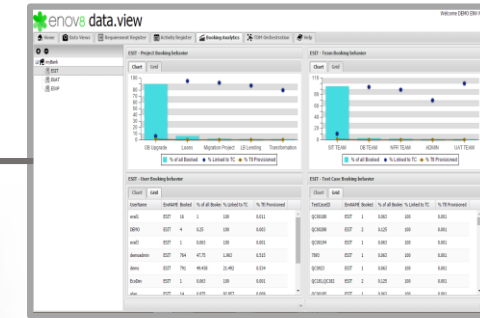
DataView supports all major data sources including:

- ✓ RDBMS – Including Oracle, MSSQL & DB2 & MYSQL
- ✓ Legacy - Mainframe IMS, VSAM
- ✓ Flat File – XML, CSV ,
- ✓ Proprietary/Abstract e.g. Hogan, SIEBEL, SAP

Note: Other data bridges can be built on demand.



*DataView



*DataHotSpot (Admin Function)

