A Virtual Bank For Development And Testing

Lessons learned from a large synthetic test environment

Daniel Künzle
Information Technology, Global Testing Services
Credit Suisse AG
Zürich, Switzerland
daniel.kuenzle.2@credit-suisse.com

Carl Worms
Information Technology, Enabling Practices & Tools
Credit Suisse AG
Zürich, Switzerland
carl.f.worms@credit-suisse.com

This paper describes how a Virtual Bank – a development and testing environment based on a purely synthetic test data base – has been established to address the challenge of protecting sensitive business and client data, while at the same time collaborating in application development and testing across country and jurisdiction borders.

Keywords—virtual enterprises; data privacy; programming environments; software testing

I. INTRODUCTION

IT is the key enabler for Credit Suisse. It is globally organized and operates hubs in Zürich, London, New York and Singapore. Zürich – the largest hub – operates an application landscape of around 700 mostly highly complex and interlinked business applications running on a multitude of infrastructure technologies including thousands of client-server systems and a large mainframe-based backend system. Outsourcing, near- and off-shoring have become ubiquitous for application development. The banks large IT projects are almost always global endeavors.

Due to the huge complexity and size of the banks data (several thousand data bases, ten thousands of data attributes), test data was typically copied from the operational data bases. As IT staff in a development environment typically have direct access to data in the test databases, that data had to be permanently anonymized.

II. LIMITATIONS OF TEST DATA ANONYMIZATION

Several attempts of analysis aimed to leverage the existing anonymization techniques to fulfill our extended requirements for cross-border development and testing. Yet none of these was successful.

The regulator defines three sensitivity categories for data:

1. Directly identifiable data: Client names, complete addresses, e-mail addresses, phone numbers, etc.
2. Indirectly identifiable data: Client id numbers, account numbers, credit card-numbers, etc.
3. Potentially indirectly identifiable data: Date of birth, citizenship, credit limits, transactions, etc. – basically all other data related to a client

While anonymization techniques are well suited for handling categories 1 and 2, they come to their limits with category 3. This type of data may lead to the identification of a client by combining individual data items that are harmless by themselves.

Any client-related data item may or may not be relevant to identify a client - one cannot say in advance. Therefore, basically all client-related data had to be anonymized. You can easily imagine that keeping your test data consistent with such an approach is just not feasible in a complex application landscape. But since exactly the anonymization of category 3 data is an explicitly stated requirement, a different approach had to be found.

III. SOLUTION: VIRTUAL BANK WITH SYNTHETIC DATA

This approach was as straightforward as bold: If production data copies could not be used, test data had to be created from scratch. And as there isn't anybody who comprehensively understands the data structures of the application landscape in scope, data should be created in the same way as in production - by just running the respective business applications, however with artificial (i.e. synthetic) clients who have never existed in the real world. In this way, also "business" generated with such synthetic clients would be synthetic with no relation to real business. The idea of the "Virtual Bank" was born.

A. Virtual Bank principles

We established four principles for a Virtual Bank:

1) Synthetic test data principle

Synthetic test data is strictly created from scratch, not by scrambling, modifying or replacing (i.e. anonymizing) copies of production data. Therefore it is inherently impossible to track it back to real clients or business data at all. The heart of synthetic test data is the synthetic client base with accounts and safekeeping accounts – so called "CAS" data. Accounts are initially empty, i.e. balance is 0. Synthetic clients must be clearly identifiable as such and therefore follow specific rules:

- Client id numbers are within a reserved range which does not exist on the production systems
- The term "TEST" must appear in the clients name

978-1-4799-7125-1/15/$31.00 ©2015 IEEE

- Addresses must be real since this is verified in many application. However, only addresses from a list defined by the bank's Legal Department may be used. Also client names are limited by such a list.

2) Segregation principle

A Virtual Bank is an "island" for development and testing work, from where its users (developers, testers) cannot "escape" to environments which may contain sensitive data. Therefore, it is built in a segregated network environment, protected by firewalls.

3) Lean data principle

Data volumes in a Virtual Bank are kept low by only creating data which is really needed for testing. This lowers infrastructure and maintenance cost and shortens processing times for users. Based on potential number of applications and average number of regression test cases, we expected that in the long run data volume in the Virtual Bank that is below 5% of data volume in production systems.

4) Managed environment principle

With its own, fully independent data which is never "refreshed" with production copies, a Virtual Bank must be operated in many ways like a real bank. This includes end-of-day processing (EDP), keeping reference data up-to-date and ensuring that business processes are properly executed end-to-end.

B. Virtual Bank Implementation

Our Virtual Bank is called "DvSZ" (for „Development Server Zone") and consists of three different testing environments, supporting different test levels, as shown below:

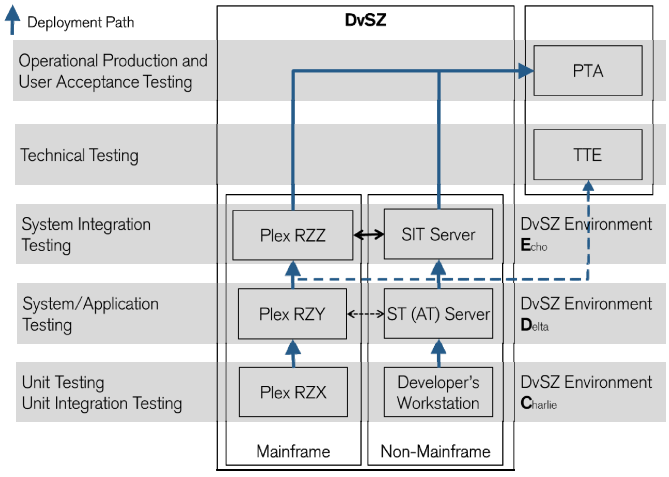

Fig. 1. DvSZ provides a separate environment for each functional test level

Environment "Echo" – for System Integration Testing (SIT) – is operated as the Virtual Bank and is such the source of the synthetic test data base. The other two environments, "Delta" – for System/Application Testing (ST) – and "Charlie" – for Unit/Unit Integration Testing (UT/UIT) – are refreshed from "Echo" after each quarterly main production release.

Each environment has its own IBM mainframe data centre instance (SYSPLEX), which represents the majority of the backend system. Outside mainframe, development and UT/UIT is done on the developers workstation. Test servers are connected to the mainframe according to the test level.

External partners outside Switzerland access DvSZ exclusively through Citrix-based virtual desktop infrastructure (VDI). This allows those companies to use their regular infrastructure without high performance requirements or Credit Suisse-specific configurations, avoid complicated installation procedures outside of Credit Suisse offices and allows Credit Suisse to centrally control desktop security parameters. This setup also avoids Credit Suisse assets (source code, documents) being physically transferred and stored outside Credit Suisse.

IV. RESULTS

Today, 450 business applications are established in the DvSZ for development and testing work. This allows approximately 1'500 people to work safely in a shared environment from different locations around the world. Compared to earlier cross border collaboration, efficiency has sky-rocketed.

The low data volume has lowered operating cost for infrastructure and shortened processing times. Alone annual mainframe CPU cost for EDP – the main CPU usage and cost factor – could be reduced by 2 Mio CHF, as less data is processed. At the same time, EDP can complete in much shorter time, leaving more time for testing and even allowing to run EDP more frequently.

The complete absence of sensitive client and business data has also allowed to operate the environment from a low-cost location. In addition to the round 75% cost savings on staff, the different time zone allows to run EDP early in the morning – before work starts in Switzerland – and extend support hours.

V. CHALLENGES

DvSZ requires creation and careful handling of test data. This is still a challenge for parts of the testing organization, who was used to have humongous volumes of production-like test data at their disposal "for free". Also, historical data, required e.g. for tax calculations or portfolio valuations, can only emerge over time. Even in a Virtual Bank living its virtual life, it takes one year of operation to get one year of data history. By regularly refreshing "Charlie" and "Delta" environments with the data from "Echo", the three environments contain the same set of test data, despite the test cases in these three environments are supposed to be different. This means that each environment contains actually more test data than really needed for testing. However, this redundancy is relatively small, compared to production volumes.

VI. OUTLOOK

Additional applications are being migrated with the aim to host all 700 business applications in Switzerland for development and testing. With the increasing user base, environment management and particularly management of the Virtual Banks synthetic test data has become more important, in order to keep it consistent in a technical sense as well as from a business point of view.