

RestFul Fault Injector

Jim Rice & Bharath Hegde
EMC Corporation

With the advent of cloud storage, REST(Representational State Transfer) is becoming the common method for access and utilization of cloud storage systems. This presentation explains a product development accelerator that can help in reducing the cycle time in development of REST based cloud storage and ISV applications by using an error injector.

A RESTful error injector helps ISV to develop reliable applications for cloud storage. This tool facilitates simulation of error scenarios by injecting REST and HTTP error codes to validate handling of those errors by the application. This presentation explains the high level approach, the implementation details in Windows, tuning to enable usage of this accelerator across various cloud storage systems and configurations to simulate different error scenarios. The details of the implementation are elaborated by taking EMC Atmos cloud storage as an example.

Learning Objectives

- ❑ Get to understand cloud storage testing.
- ❑ Usage scenarios of the error injector to reduce development time.
- ❑ Hook to Windows HTTP to enable REST error injection.
- ❑ Http responses and error codes returned by REST API.

Agenda

- ❑ Cloud Storage
- ❑ REST
- ❑ RESTful Error Injector
 - ❑ Architecture
 - ❑ Implementation
- ❑ Conclusion
- ❑ Acknowledgements
- ❑ Q & A

Cloud Storage

OVER THE INTERNET

SHARED

PAY AS YOU GO

SELF-SERVICE

USE ONLY WHAT YOU NEED

SCALABLE

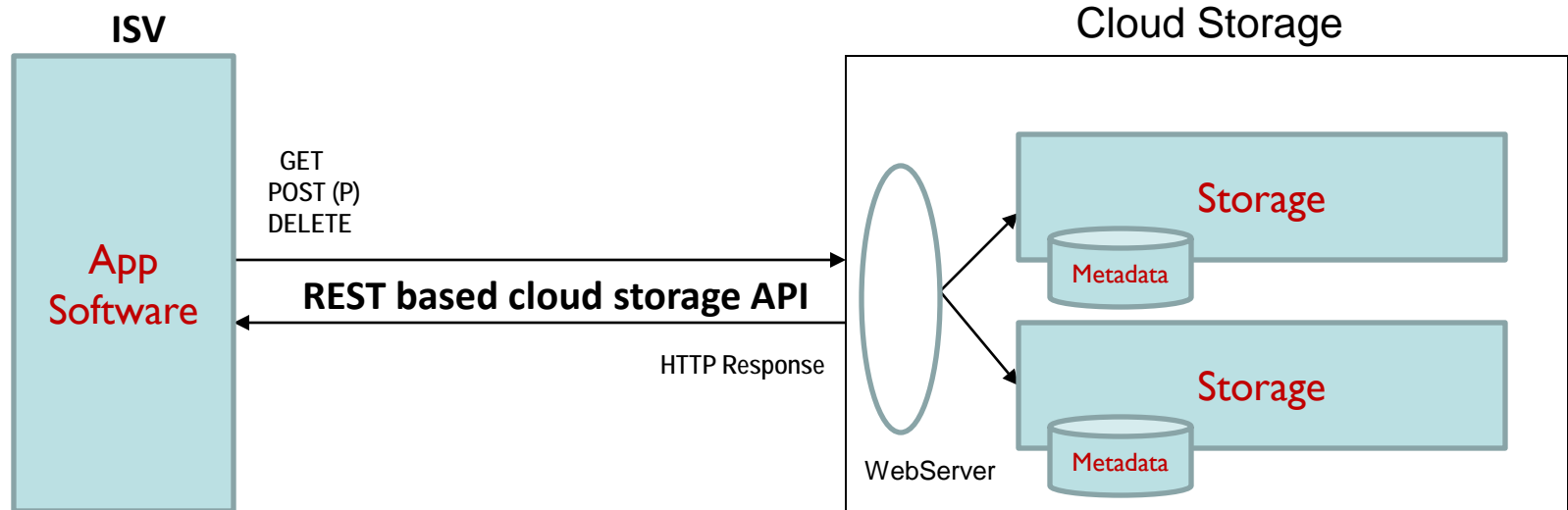
...On-Demand!

- ❑ Acronym for: Representational State Transfer.
- ❑ REST is *an architecture style* for designing networked applications. Systems following these principles are referred to as 'RESTful'.
- ❑ Simple HTTP is used to make call between machines.
- ❑ Resources are URL-Addressable. Easy utilizing HTTP Methods: GET, POST, PUT, DELETE.
- ❑ Any simple interface which transmits domain-specific data over HTTP without an additional messaging layer such as SOAP or session tracking via HTTP cookies.
- ❑ A popular method used for accessing data in the cloud.

What does a Cloud Storage API do for an ISV?

- ❑ A cloud storage API is a method for accessing a cloud storage system.
- ❑ Allows for native communication between the software application and the cloud storage provider.
- ❑ Eliminates the need for the ISV's application to depend on an appliance for movement of data to cloud.
- ❑ Allows ISV greater control on managing data stored in cloud.

RESTful Cloud Storage



- ❑ An ISV uses REST based API to create and manipulate object data(such as files) and its metadata(such as tags) from the application.
- ❑ The REST Web service uses HTTP methods like POST, DELETE, PUT and GET to create, delete, update and read objects and metadata.
- ❑ The Web services APIs could support an object interface and a file-system-like namespace interface.
- ❑ Application could associate metadata with objects that get stored in the cloud storage. This metadata can be used to trigger policies based on application goals.

RESTful Error Handling

Generic methodology that a REST-based Web Service can follow to return error messages to client application:

- ❑ HTTP Error Codes – Propagate web service error messages via standard HTTP error codes.
- ❑ Extended HTTP headers – Specify an application specific error code within a separate HTTP Header.
- ❑ Return an XML error document – Always return HTTP status code of 200, but also include an XML document containing application specific error message.

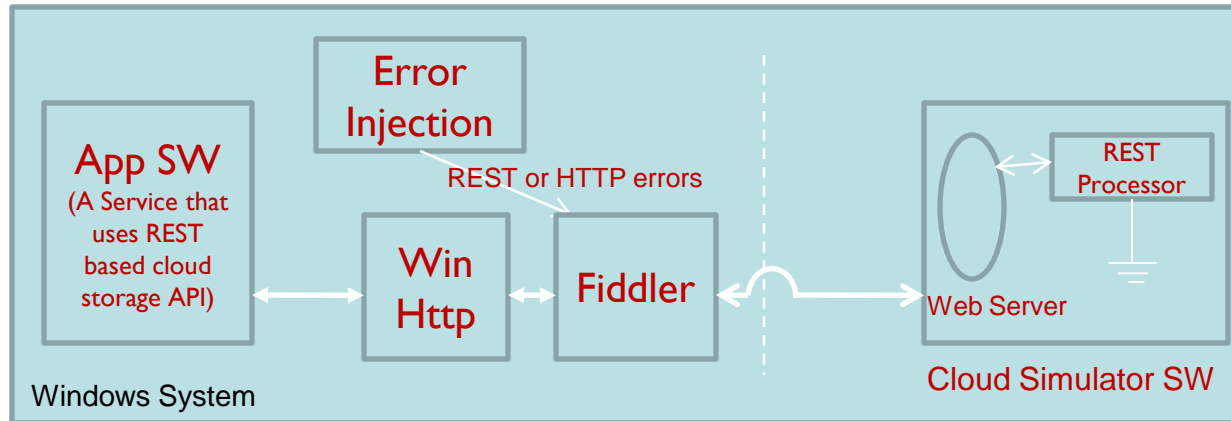
Many REST based cloud storage API use a combination of the above to handle errors. A typical error methodology used is:

- ❑ Return HTTP status codes for problems specific to HTTP.
- ❑ When an error occurs, return an XML document detailing the error.
- ❑ The error document contains both an error code that can be machine processed and a human readable error message.

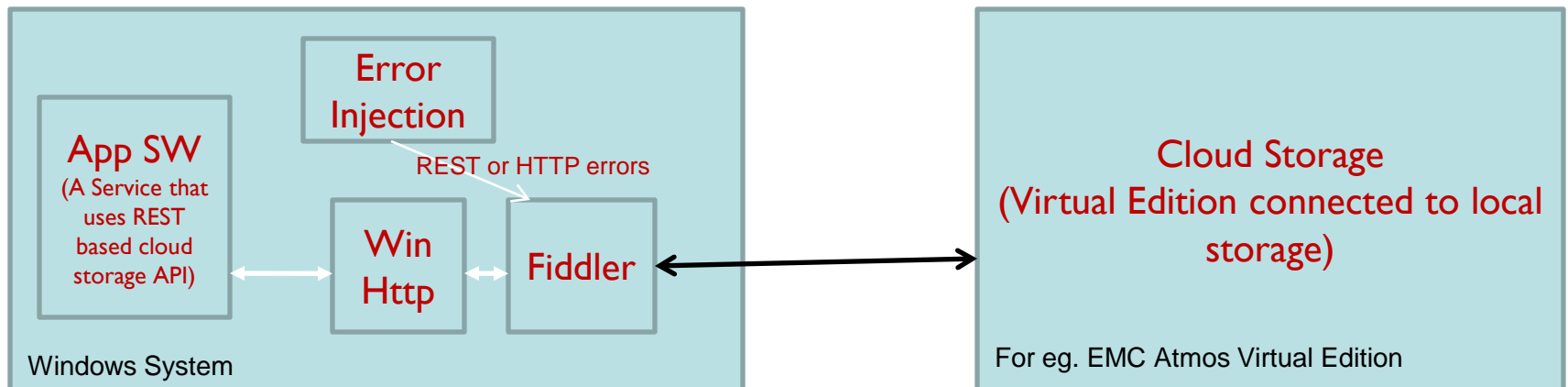
Why RestFul Error Injector?

- ❑ Enables application testing to be done without dedicated cloud infrastructure.
- ❑ Simulator to fabricate REST/HTTP requests and responses to inject error scenarios, rather than creating environments to get these errors.
- ❑ Can be used for testing with multiple cloud storage vendors.
- ❑ 3rd Party HTTP Request Builder can be used to build error packets.

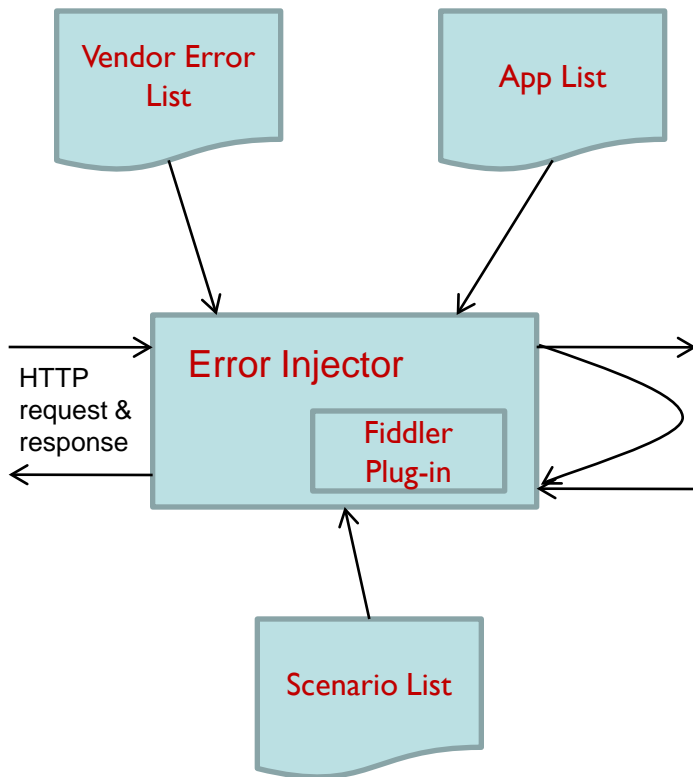
Error Injector - Architecture



OR



Error Injector - Implementation



- ❑ The core error injector is extension of Microsoft [Fiddler](#) HTTP debugger.
- ❑ Fiddler is extensible using the .NET Framework. There are two primary mechanisms for extending Fiddler: Custom Rules and Inspectors.
- ❑ Fiddler allows to add plug-in **Inspector** objects written in any .NET language. **RequestInspectors** and **ResponseInspectors** provide a format-specific or an otherwise specialized view of the HTTP request or response.
- ❑ Three configuration files are inputs to the error injector:
 - ❑ Vendor Error List - Vendor specific error code along with error message and mapping HTTP error code.
 - ❑ Scenario List – On various scenarios to inject errors.
 - ❑ App List – Client(App SW)-Webservice(Cloud storage) pair to be monitored for error injection.
- ❑ Requests go through the error injector before going out to the cloud storage device. Error(s) can be injected to the request using request inspectors.
- ❑ On receiving response, the response can be modified as an error response using response inspectors.

Error Injector – Key Features

- ❑ Capability to inject REST and/or HTTP error codes.
- ❑ Dynamic configuration of the tool based on the cloud storage vendor error list.
- ❑ Option to use a cloud storage simulator that can drop data packets or persist data packets to a virtual edition software connected to storage.
- ❑ Facility to configure the type of HTTP/REST errors. For eg:
 - ❑ Every 5th GET request inject error in vendor specific field.
 - ❑ Every 7th DELETE response needs to be time-out.
 - ❑ Every 3rd, 6th, 9th etc response for “Reading an object” REST request should be returned with a REST error code.
 - ❑ Delay response by x seconds for every 10th “Getting an ACL” REST request.
- ❑ Capability to inject error before and after sending a request/response on the wire.

Benefits of Simulation

- ❑ ISV can develop robust applications without actual cloud infrastructure, improving time to market. Avoids, lead time to procure cloud storage hardware or testing delays due to latency in talking to remote cloud infrastructure many miles away in cloud storage vendor campus.
- ❑ Cheap: No special hardware required to setup Cloud.
- ❑ Facilitates automation to inject all types of error scenarios enhancing quality and reducing cycle time for testing.
- ❑ Easily deployable in development environment for unit testing.
- ❑ Improved testing in various error scenarios ensuring better code and test coverage.

Summary

- ❑ Cloud storage is emerging as an important class of storage system.
- ❑ Cloud storage predominantly uses REST API.
- ❑ This error injector is a tool to simulate HTTP errors in ISV applications which work with cloud storage.

References

1. REST - http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
2. EMC Atmos Programmers Guide - <https://community.emc.com/docs/DOC-3481>
3. Fiddler - <http://msdn.microsoft.com/en-us/library/Bb250446.aspx>
4. RESTful error handling - http://www.oreillynet.com/onlamp/blog/2003/12/restful_error_handling.html
5. Comparison of fault and error injection - http://citeseerx.ist.psu.edu/viewdoc/download%3Fdoi%3D10.1.1.69.8102%26rep%3Drep1%26type%3Dpdf&sa=U&ei=BFUTsTGKsTqmAWty8kB&ved=0CBgQFjAD&usg=AFQjCNH6qTQbDsxtXMAL91Tj_mhvUbnJzQ

Acknowledgement

We are thankful to

Anil Degwekar (anil.degwekar@emc.com)

- Architect, EMC Corporation

Vinodraj Daniel (vinodraj.daniel@emc.com)

- Engineering Manager, EMC Corporation

for their ideas and suggestions.

Questions?

You can email queries to

jim.rice@emc.com