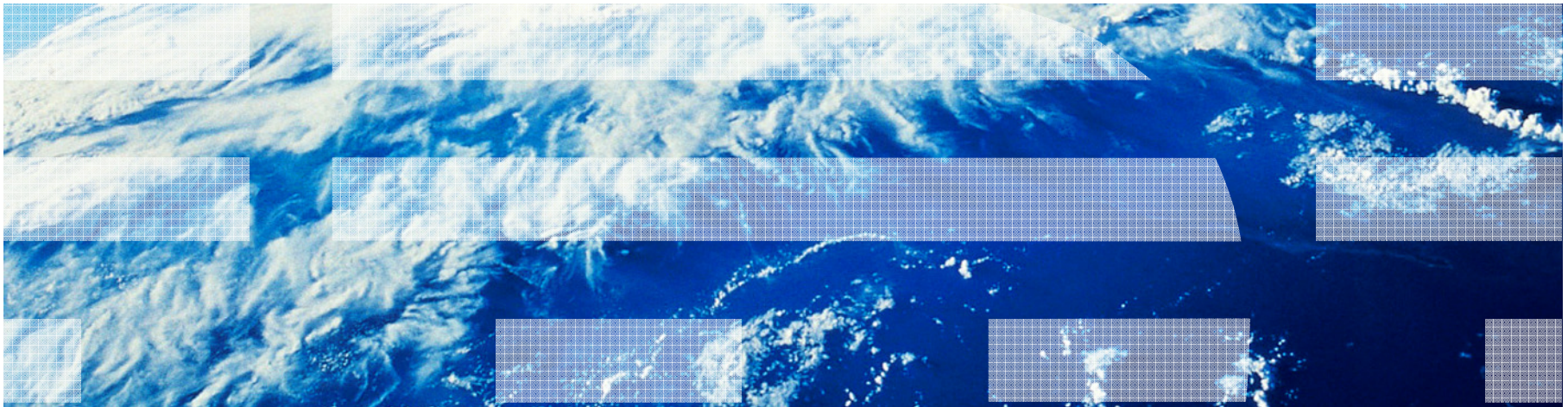


Storlet Engine for Executing Biomedical Processes within the Storage System

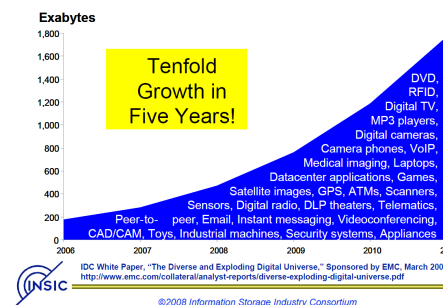
Simona Rabinovici-Cohen, Ealan Henis, John Marberg, Kenneth Nagin

{simona, ealan, marberg, nagin}@il.ibm.com



Emerging Trends Motivate Storlets

- **“Data is the new Oil”**
 - In its raw form, oil has little value
 - Once processed and refined, it helps power the world
- **Data deluge of biomedical data**
 - Medical images, genomic sequencers, videos, etc.
 - Multiple large objects created by various devices
- **Object storage for content depots generally:**
 - Utilizes large bandwidth to serve **big data over the WAN**
 - Uses commoditized of-the-shelf hardware with **under utilized CPUs**



- ➔ **Process and refine the data where it is stored**
- Create a computational object storage with **storlets**

Storlets

- The concept of storlets was first introduced in our MSST 2007 paper to offload data-intensive computations to the long term archival storage

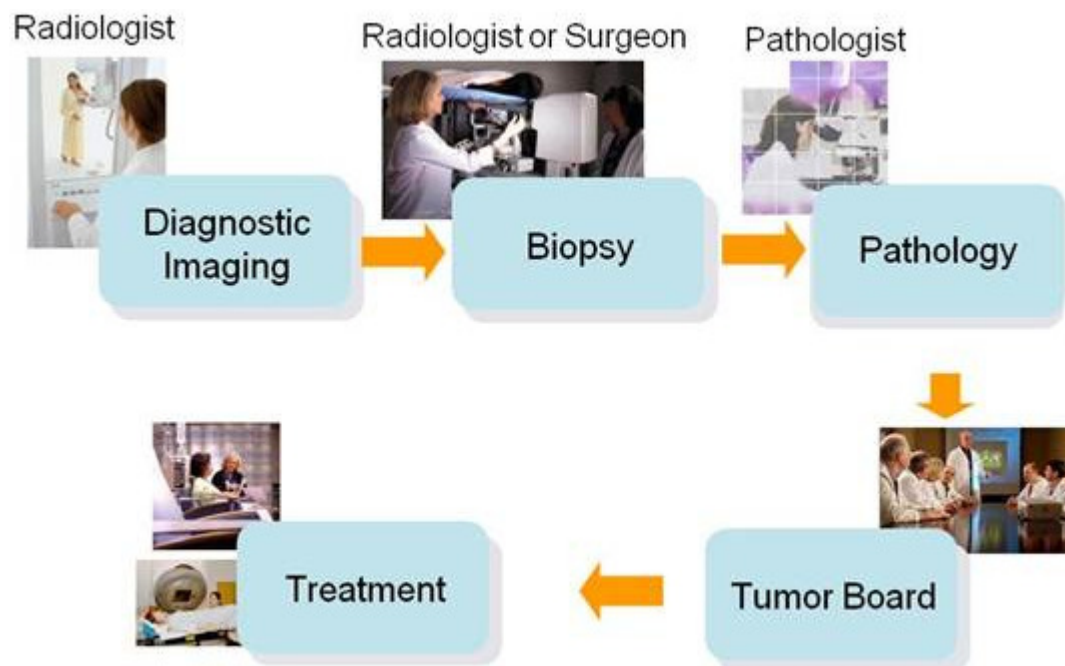
What is a storlet?

- Dynamically uploadable computational module running within the storage close to the data
- Executed in a sandbox when not trusted
- Analog to stored procedures in databases

The Storlet Engine mechanism:

- Adds flexibility to the storage
- Makes the storage extensible - makes the storage a platform
- Transforms the storage from “keeping data” to “producing value from the data”

Pathology Department Use Case

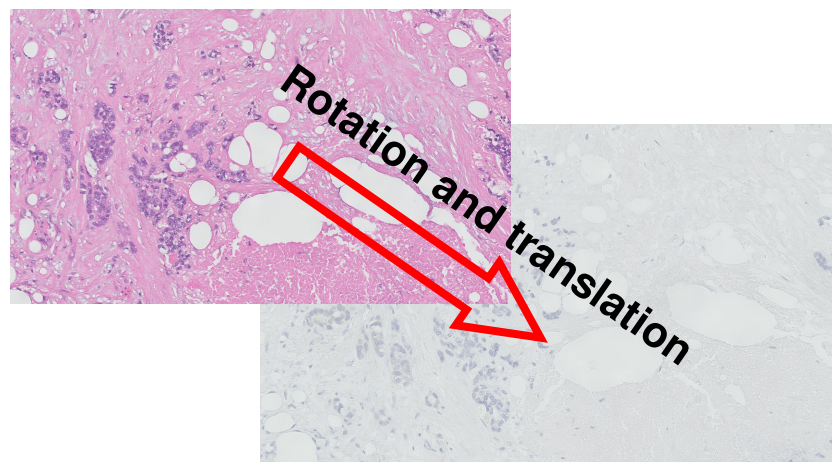


* image courtesy of Philips

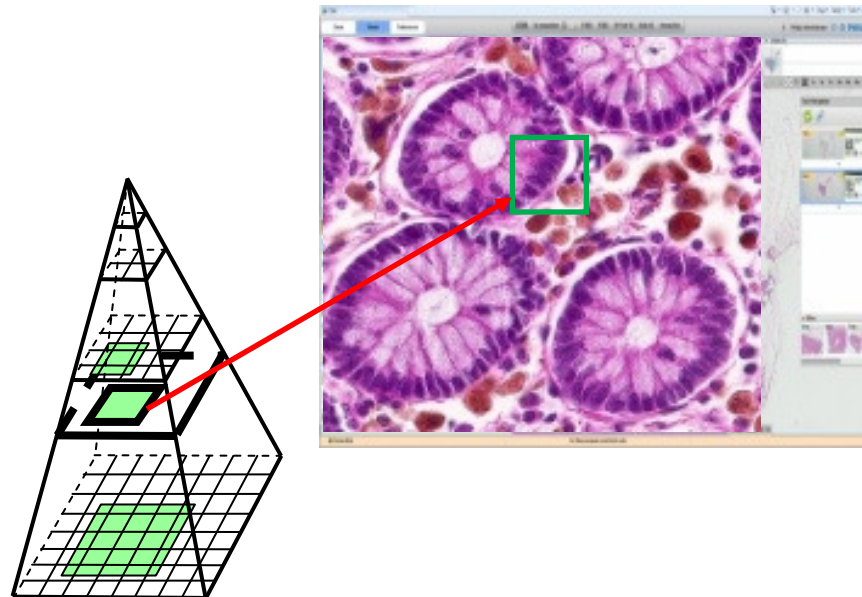
- A **tissue block** is given to the Pathologist
- Very thin **slices** can be cut off the tissue block and mounted on glass slides
- The thin slices are stained with different stain, e.g. HER2, PR, ER, etc.
- Pathology **images** are created for the stained slices
- Large images can be 200K x 200K which can consume 5-10 GB

ROI Extraction Business Process

Aligning images via rotation and translation

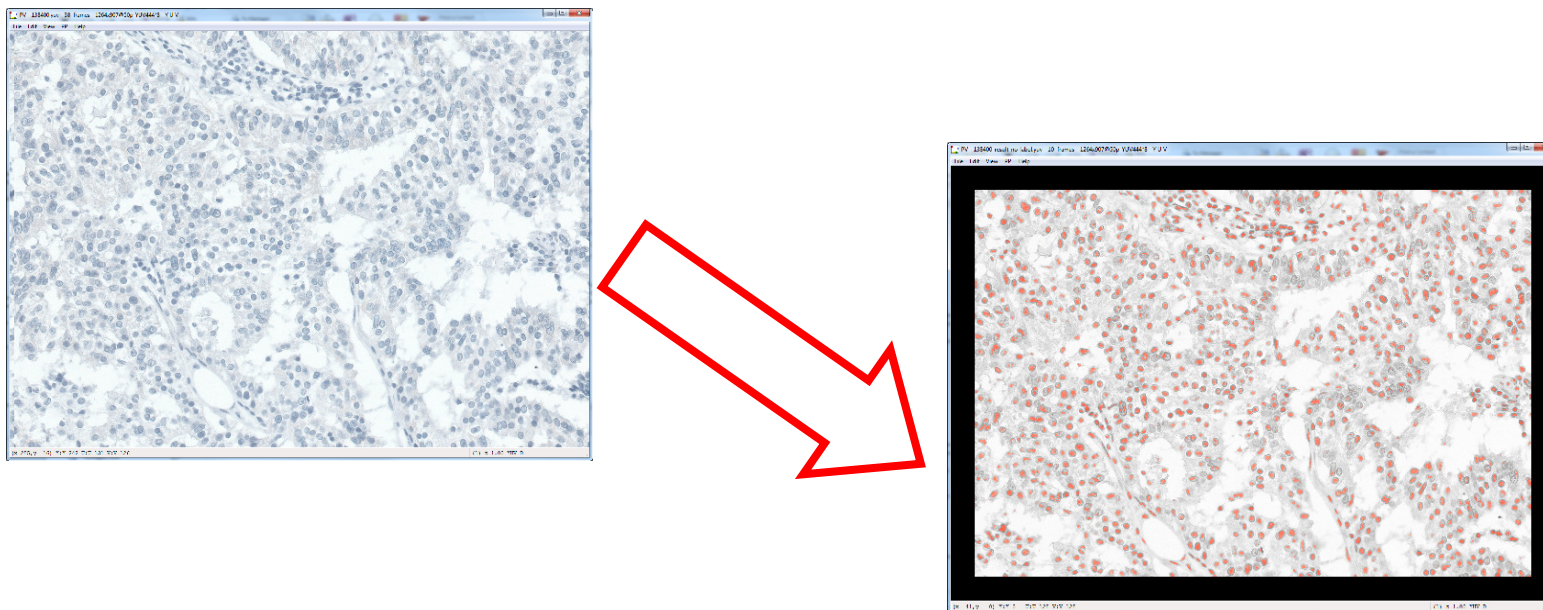


Extracting subset of the images and transforming to a standard format



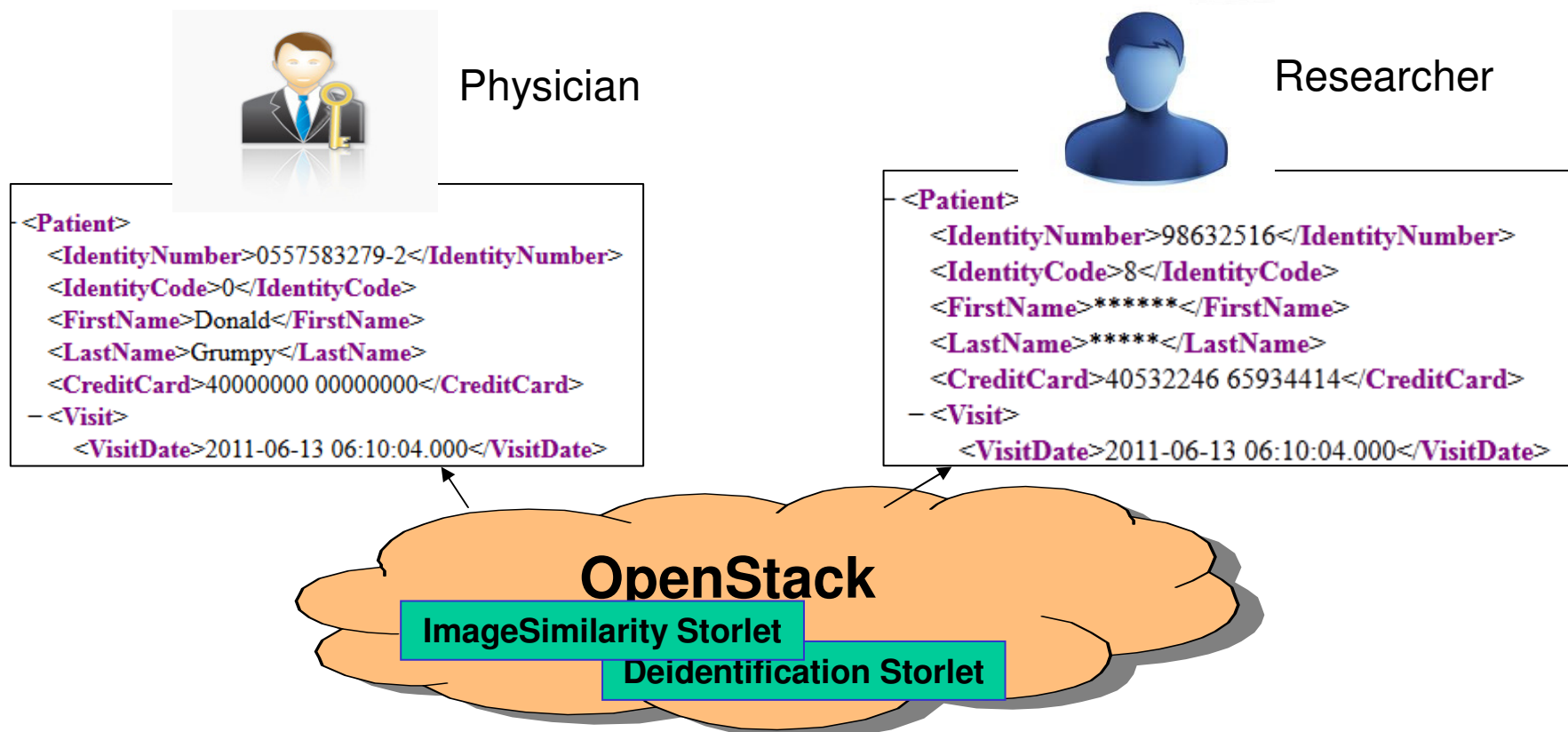
* images courtesy of Philips

Cell Detection Business Process



- Cell detection requires pattern search/feature extraction and analysis of pathology images
- This is very important for Pathologists because it provides a quick overview of the disease cells, based on which diagnosis can be made
- The oncologist machine may be too weak to execute this module
- With storlets technology, the Cell Detection Storlet will run in the cluster of machines within the storage

Cohort Identification Business Process

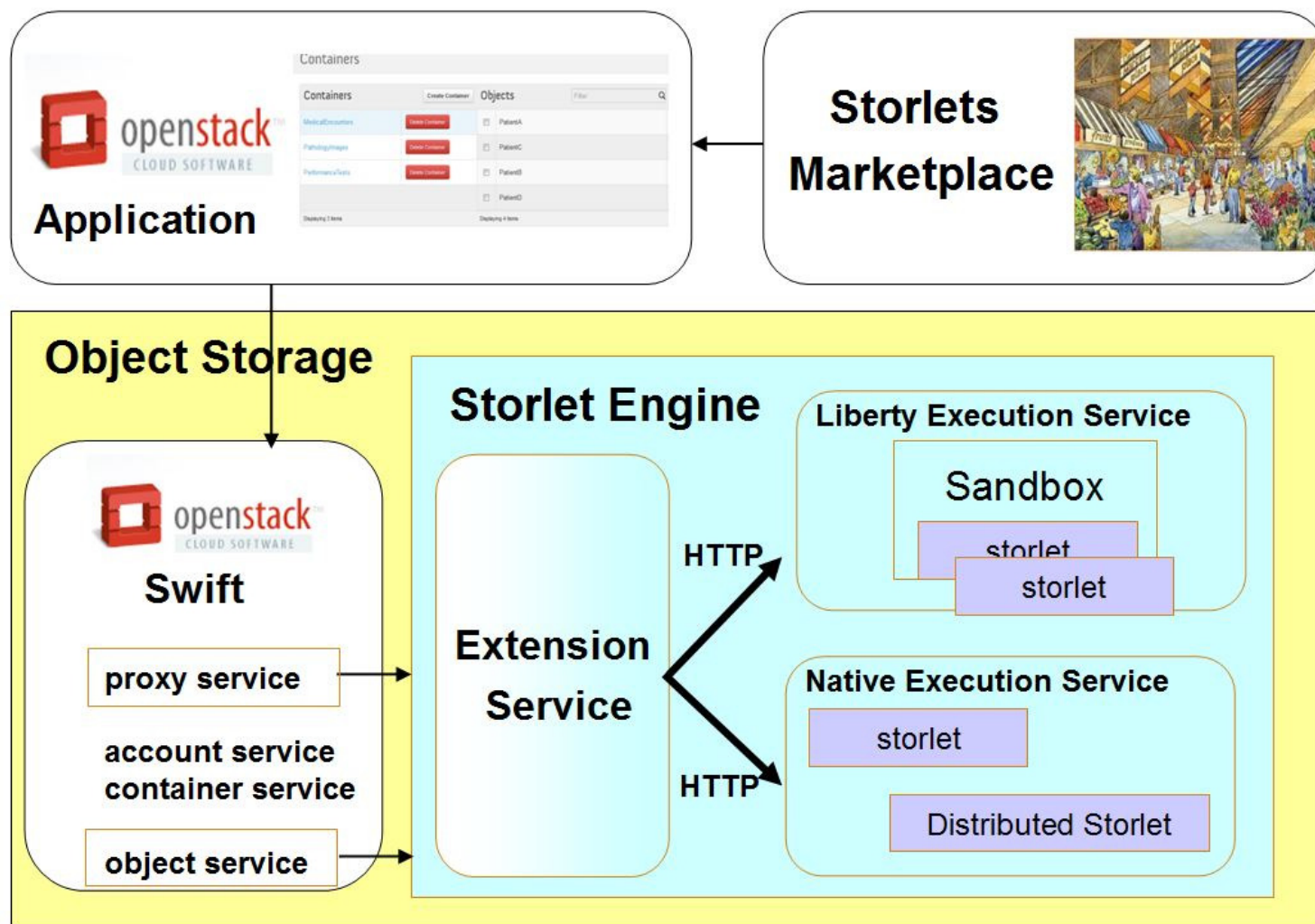


- A researcher would like to get a cohort of records and images with similar features from as multiple patients as possible
- This is more secure than the traditional method as it spares the need to move clear data to the staging system
- Helps comply with HIPAA

Client Value for Using Storlets

- **Reduce bandwidth** – reduce the number of bytes transferred over the WAN
e.g. Analytics storlet
- **Enhance security** – reduce exposure of sensitive data
e.g. De-identification storlet
- **Save costs** – consolidate generic functions that can be used by many applications while saving infrastructure at the client side
e.g. Curation storlet
- **Support compliance** – monitor and document the changes to the objects and improve provenance tracking
e.g. Transformation storlet

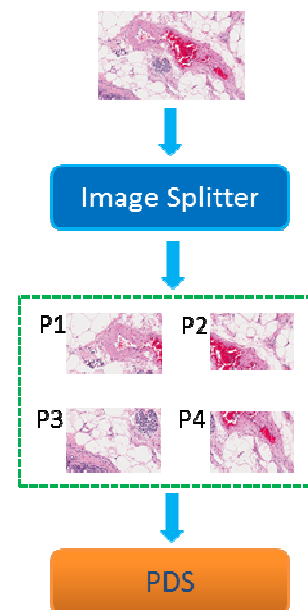
Storlet Engine Architecture



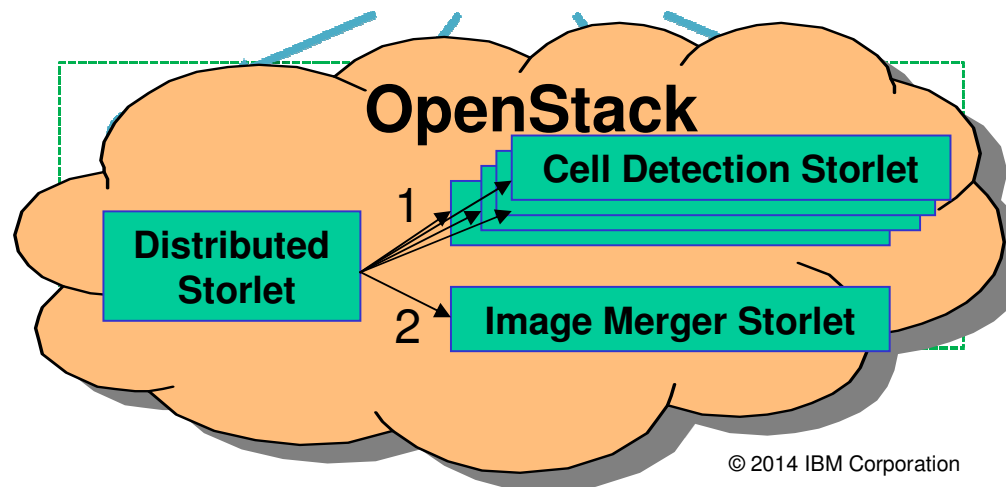
* The Native Execution Service can execute docker-based storlets

Distributed Storlet and Cell Detection

- The compound distributed storlet input includes:
 - list of data objects to work on
 - split storlet e.g. CellsDetectonStorlet
 - merge storlet e.g. ImageMergerStorlet
- The distributed storlet initiates multiple split storlets on each data object at its residence. Afterwards, it calls the merge storlet to summarize the results.
- The distributed storlet is a service in the Storlet Engine suitable for analytics processes.



- **Distributed Storlet and Image Merger Storlet run in Proxy node**
- **The multiple Cell Detection Storlets run in Object nodes**



Rules Mechanism

- Enables automatic conditional invocation of storlets
 - Explicit storlet activation overrides implicit activation
 - Rules kept as per tenant editable object, with specified access control
 - Configured by tenant, user, role, container, object, content_type
 - Wildcards (“*”) allowed in a rule (high flexibility)
 - The first rule that matches the input is activated – prioritized list of rules
- Examples:
 - De-Identification (per Role)
 - Transformation (per Content Type)
 - Fixity (per Container)

Security Model

Requirements

- Restrict and control the execution capabilities of storlets (per sandbox type)
- Isolate and protect storlets of different tenants from each other (multi-tenancy)
- Authorize and verify access from storlets to object storage

Solution Features

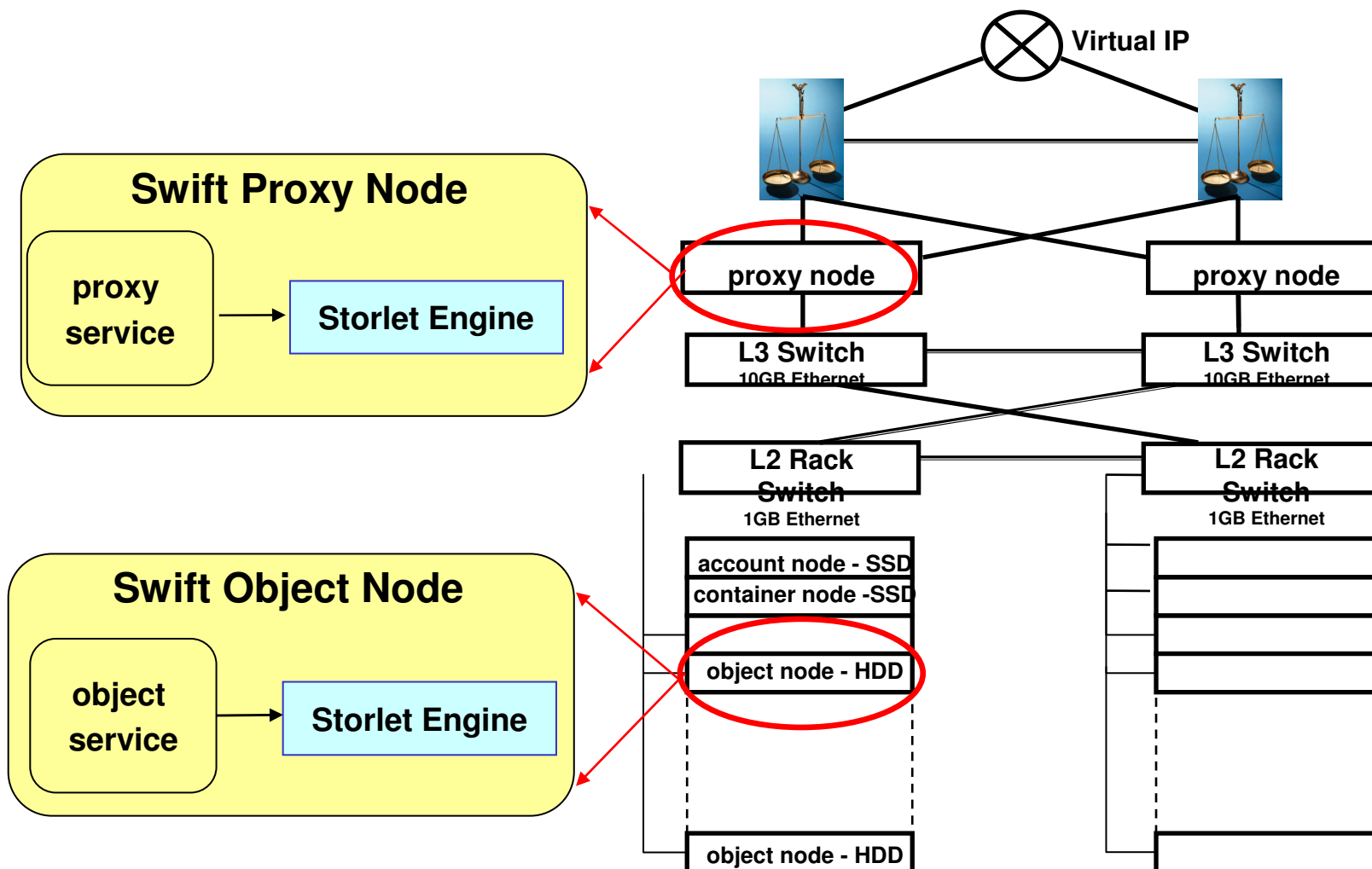
- Use a separate web server for each combination of [tenant, sandbox]
- Dispatch each storlet request to the server of the given tenant and sandbox
- Leverage Linux capabilities for protection and isolation of individual web servers
 - Each server is associated with dedicated unique Linux uid/gid and port number
 - Server and all its storlets run in a single process, deployed under the unique uid/gid
 - Server listens on its unique port number
 - Sensitive files have the server's unique uid/gid permissions
 - Communication to target ports/hosts is filtered by iptables using uid and gid rules
- Storlets use tokens to access the object storage (supplied as request parameters)

Sandbox Types

- A storlet may be associated with different trust levels:
 - Storlet can be written by a system administrator; written by a user; or bought as part of a third-party package or downloaded from some site.
 - Storlet execution can be initiated at different privilege levels: by an administrator; by a privileged user; or by a regular user.
 - Based on the source of the storlet, the initiator, and the storlet functionality, a certain level of trust should be associated with the storlet.

- Storlets can run in various types of sandboxes to associate different levels of trust with different storlets:
 - **Admin Sandbox** – the storlet can do all operations.
 - **User Sandbox** - the storlet is restricted and cannot perform operations such as write to the filesystem, open a socket or access another host directly, spawn threads, etc.

Storlet Engine Deployment in OpenStack Swift

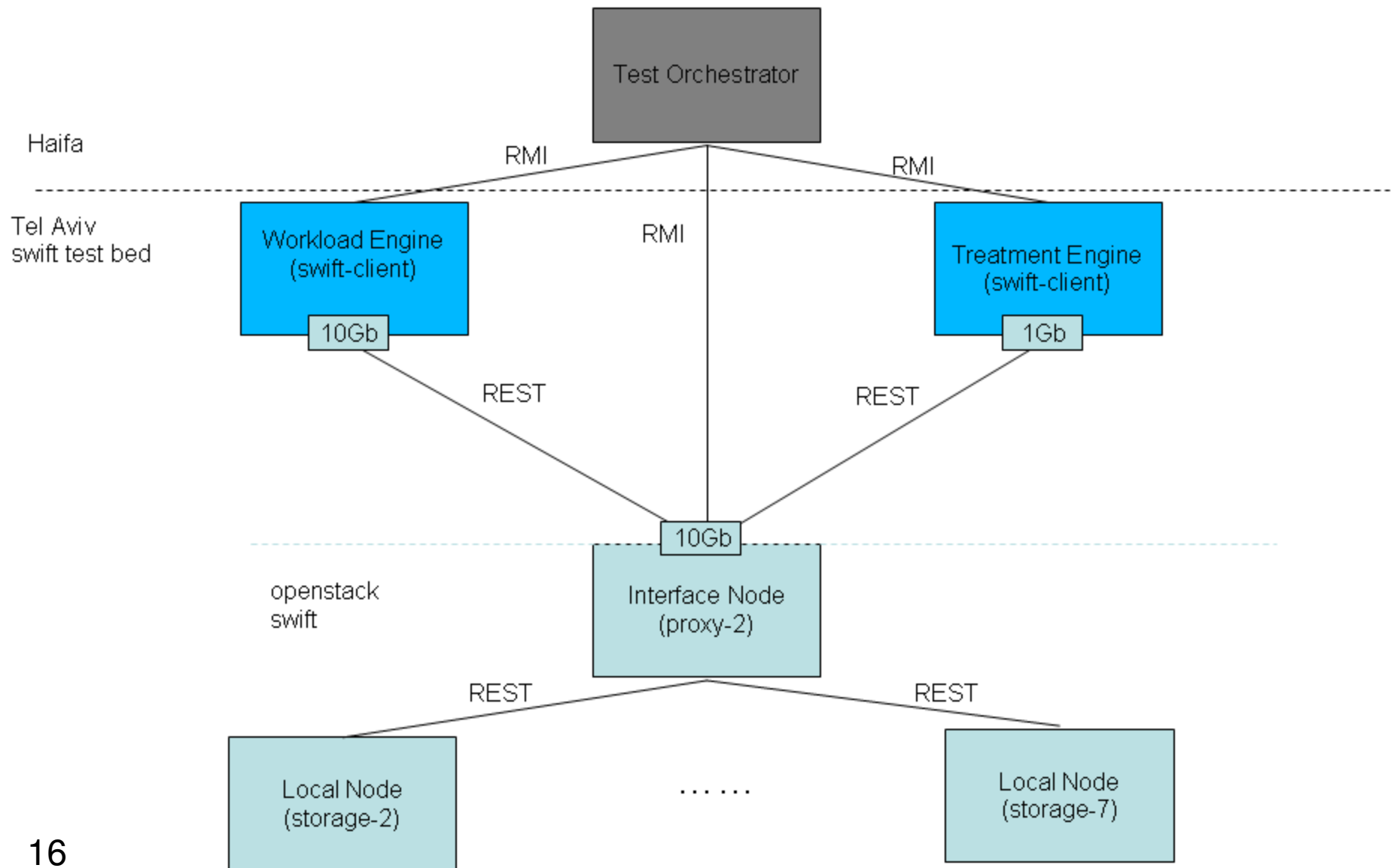


Performance Study: Goals

Answer these questions:

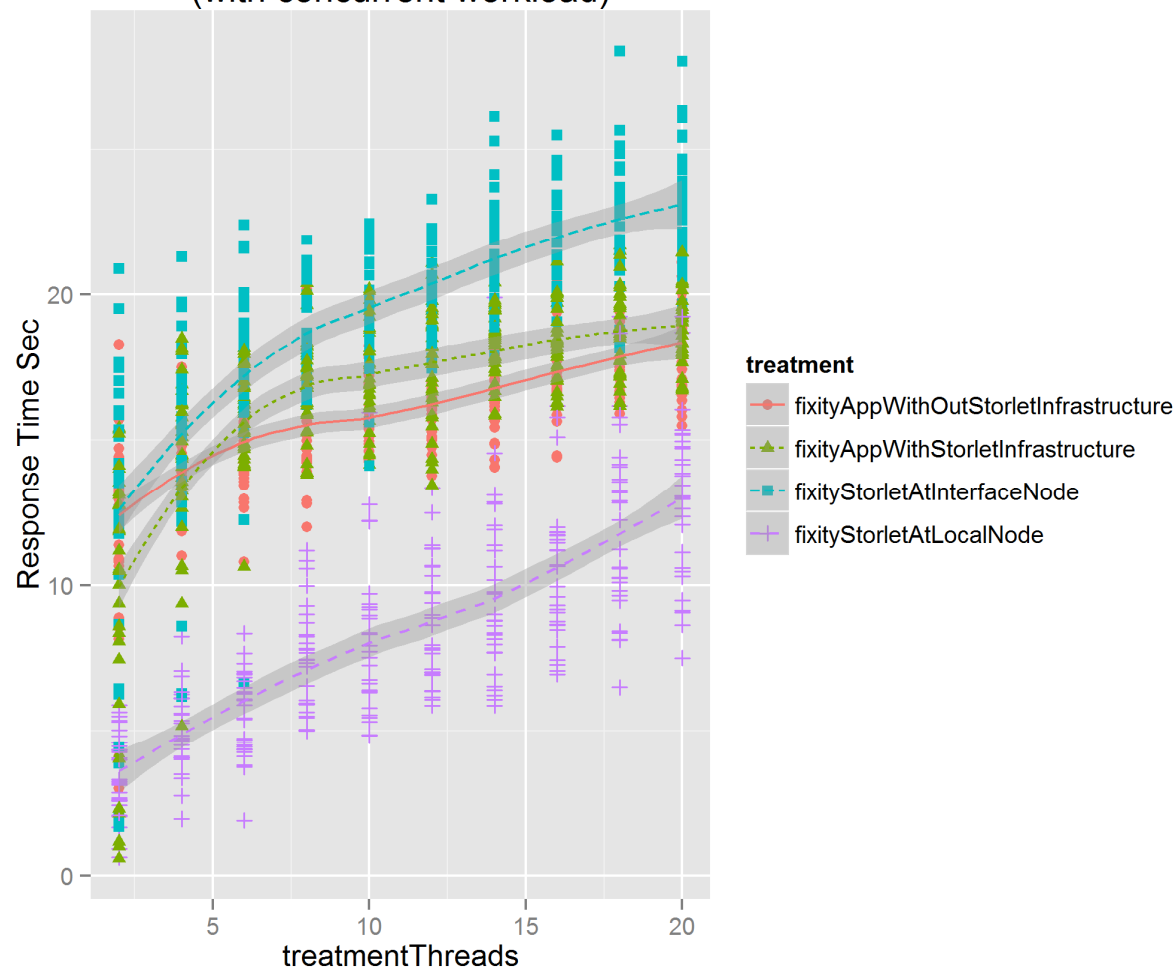
- What performance benefits can be derived by wrapping a function as a Storlet?
- How Storlets affect system performance?
- Performance for Storlets on local storage nodes versus Storlets on interface node?
- What host resources are most affected by Storlet?
- Do the performance issues change for private cloud (LAN) or public cloud (WAN)?

Test setup



Fixity Test Results Treatment Response Time

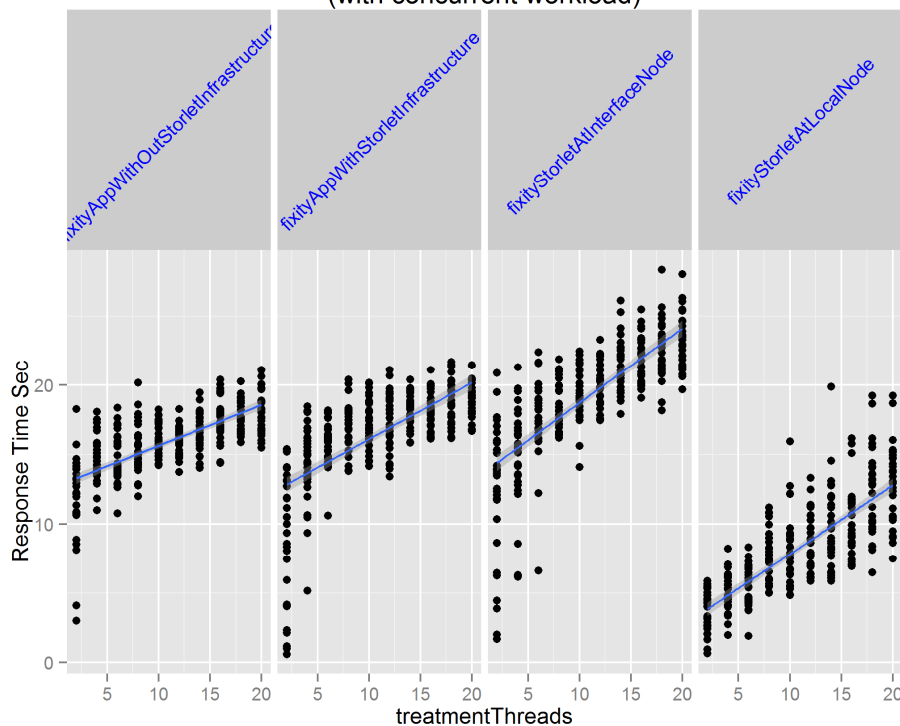
Treatment Content Type: MB_5-100by5
 Treatment to Interface Node
 Avg. Latency: 0.091
 (with concurrent workload)



Fixity Test Results

Treatment Response Time (fitted linear models)

Treatment Content Type: MB_5-100by5
 Treatment to Interface Node
 Avg. Latency: 0.091
 (with concurrent workload)



	Intercept	Slope	R.squared	std-error	correlation
fixityAppWithOutStorletInfrastructure	13	0.29	0.479	0.017733	0.692
fixityStorletAtInterfaceNode	13	0.54	0.557	0.02784	0.746
fixityStorletAtLocalNode	2.9	0.5	0.609	0.023062	0.781
fixityAppWithStorletInfrastructure	12	0.41	0.443	0.026465	0.666

Conclusions

- Introduced storlets for biomedical processes e.g.
 - ROI Extraction
 - Cell Detection
 - Cohort Identification
- Presented the Storlet Engine, an environment supporting computations within storage system
 - Architecture and key features of the system
 - Rules mechanism for automatic conditional invocation of storlets
 - Security model that supports storlets multi-tenancy and various types of sandboxes
 - Performance study of the Storlet Engine prototype
- Implemented a Storlet Engine prototype for OpenStack Object Storage (code-named Swift) in the context of the ENSURE and ForgetIT EU projects

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 270000 and under grant agreement 600826.

The logo for the ENSURE project, featuring the word "ENSURE" in a bold, sans-serif font. The letter "S" is stylized with a circular graphic element inside it.

<http://www.ensure-fp7.eu/>

The logo for the ForgetIT project, featuring a stylized icon of four overlapping squares (two white, two light green) to the left of the word "ForgetIT" in a sans-serif font.

<http://www.forgetit-project.eu/>

DEMO

ROI Extraction: Trigger Storlet Selection



The screenshot shows the OpenStack Containers dashboard. On the left is a sidebar with the OpenStack logo and navigation menus for 'Project' (demo), 'Manage Compute' (Overview, Instances, Volumes, Images & Snapshots, Access & Security), and 'Object Store' (Containers). The main content area is titled 'Containers' and is split into two panes: 'Containers' and 'Objects'. The 'Containers' pane lists three containers: 'MedicalEncounters', 'PathologyImages', and 'PerformanceTests', each with a 'Delete Container' button. The 'Objects' pane shows two objects: 'Image1Aip' (251.1 MB) and 'Image4Aip' (20.4 MB). A context menu is open over the 'Image4Aip' object, showing options: 'Copy', 'Delete Object', and 'Trigger Storlet'. A green arrow points to the 'Trigger Storlet' option.

Container	Object	Size	Actions
MedicalEncounters			Delete Container
PathologyImages	Image1Aip	251.1 MB	Delete Container, Download
PerformanceTests	Image4Aip	20.4 MB	Delete Container, Download, Copy, Delete Object, Trigger Storlet

Find: [] Next Previous Highlight all Match case

Trigger Storlet

Trigger Storlet On: Image4Aip

Sandbox Type
AdminSandbox

Storlet Name
ImageServiceStorlet

Storlet Params
0,0,4,600,600

Description:
A storlet is a restricted computational module executed within the storage close to the data. Triggering a storlet initiates execution of that computational module on the given object.

Cancel Trigger Storlet

openstack
DASHBOARD

Project

CURRENT PROJECT
demo

Manage Compute

- Overview
- Instances
- Volumes
- Images & Snapshots
- Access & Security

Object Store

- Containers

ROI Extraction: Trigger Storlet Result



openstack

DASHBOARD

Project

CURRENT PROJECT
demo

Manage Compute

Overview

Instances

Volumes

Images & Snapshots

Access & Security

Object Store

Containers

Trigger Storlet

Trigger Storlet On: Image4Aip

Sandbox Type

AdminSandbox

Description:

A storlet is a restricted computational module executed within the storage close to the data.

Storlet Name

ImageServices

Storlet Params

0,0,4,600,600

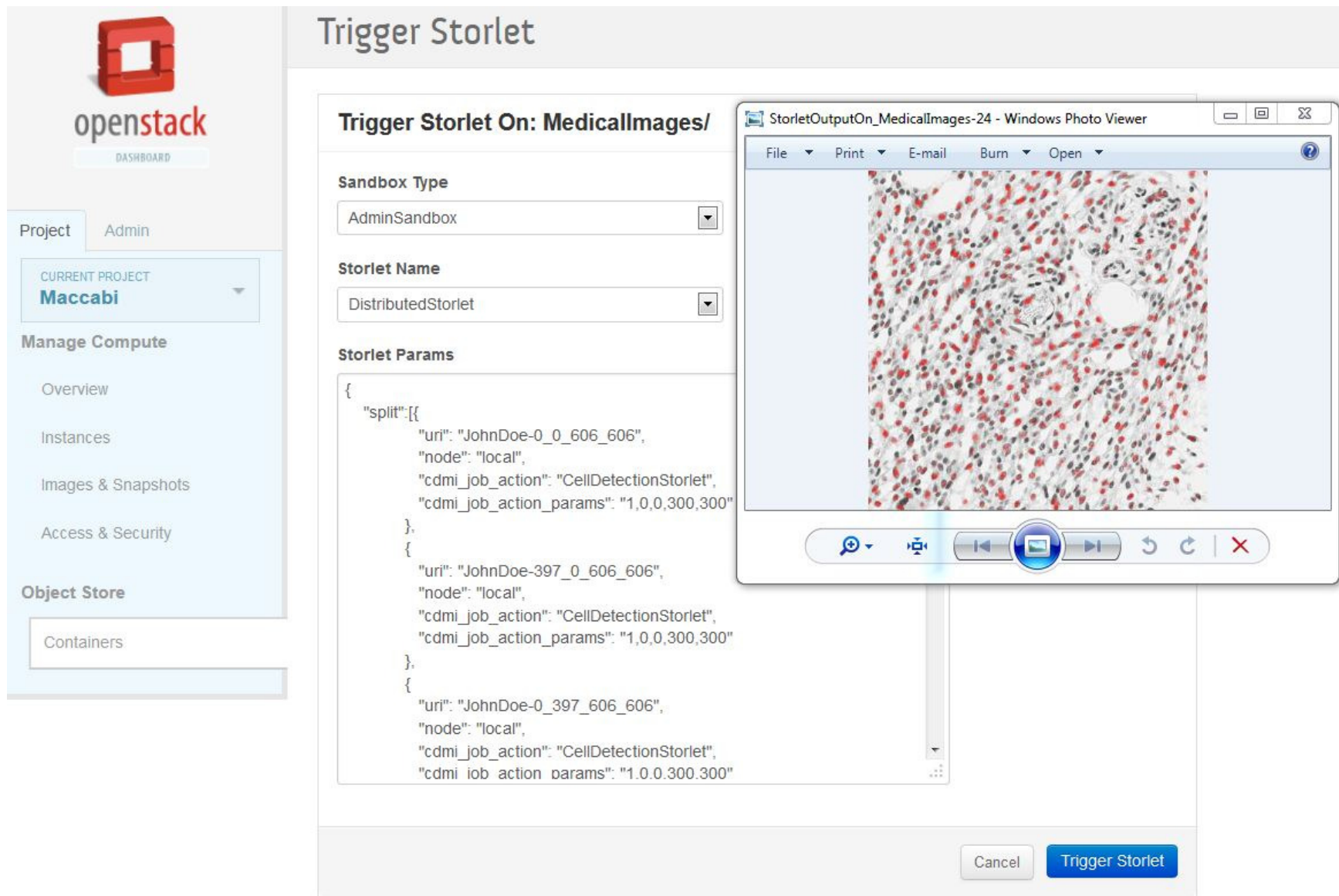
Trigger Storlet

ROI Extraction: Storlet Result on Mobile



The image shows a composite view of an OpenStack dashboard and a mobile device. On the left is the OpenStack dashboard sidebar with the logo and navigation menu. The main content area is titled "Trigger Storlet" and contains configuration fields for "Trigger Storlet On: Image4Aip", "Sandbox Type" (AdminSandbox), "Storlet Name" (ImageServices), and "Storlet Params" (0,0,4,600,600). A window titled "StorletOutputOn_Image4Aip-18.png" displays a histology image. On the right, a mobile device screen shows the same histology image with a search bar at the top containing "pds-stack1/nova/contain" and a "Search" button. The mobile interface includes a status bar with "16:07" and "92%" battery, and a bottom navigation bar with icons for back, forward, search, and other actions.

Cell Detection: Distributed Storlet Execution



Trigger Storlet

Trigger Storlet On: MedicalImages/

Sandbox Type
AdminSandbox

Storlet Name
DistributedStorlet

Storlet Params

```
{  
  "split": [{  
    "uri": "JohnDoe-0_0_606_606",  
    "node": "local",  
    "cdmi_job_action": "CellDetectionStorlet",  
    "cdmi_job_action_params": "1,0,0,300,300"  
  },  
  {  
    "uri": "JohnDoe-397_0_606_606",  
    "node": "local",  
    "cdmi_job_action": "CellDetectionStorlet",  
    "cdmi_job_action_params": "1,0,0,300,300"  
  },  
  {  
    "uri": "JohnDoe-0_397_606_606",  
    "node": "local",  
    "cdmi_job_action": "CellDetectionStorlet",  
    "cdmi iob action params": "1.0.0.300.300"  
  }  
]
```

Windows Photo Viewer: StorletOutputOn_MedicalImages-24

File | Print | E-mail | Burn | Open

Cell Detection Results (Histology Image)

Cancel | **Trigger Storlet**

Integrity Check: Trigger Fixity Storlet



The screenshot shows the OpenStack dashboard interface for configuring a Trigger Storlet. The main content area is titled "Trigger Storlet" and shows the configuration for "RoseFlowers.png".

- Sandbox Type:** AdminSandbox
- Storlet Name:** FixityStorlet
- Storlet Params:** MD5,SHA256

A description of a storlet is provided: "A storlet is a restricted computational module executed within the storage close to the data. Triggering a storlet initiates execution of that computational module on the given object." At the bottom right, there are "Cancel" and "Trigger Storlet" buttons.

Integrity Check: Fixity Storlet Result



The screenshot shows the OpenStack dashboard interface. The main content area is titled "Trigger Storlet" and displays configuration for a storlet named "FixityStorlet".

Trigger Storlet On: RoseFlowers.png

Sandbox Type: AdminSandbox

Storlet Name: FixityStorlet

Storlet Params: MD5,SHA256

Description: A storlet is a restricted computational module executed within the storage close to the data. Triggering a storlet initiates execution of that computational module on the given object.

The XML output for the fixity check is shown below:

```
<fixities>
  - <fixity>
    <fixityModule>MD5</fixityModule>
    <fixityValue>15C618567C86915F691D3C91AB0B4F</fixityValue>
  </fixity>
  - <fixity>
    <fixityModule>SHA256</fixityModule>
    <fixityValue>9D2F46D48B042EAD361EC4E73FF34CC0EA49E4F9F6348A63562411E8F42C554A</fixityValue>
  </fixity>
</fixities>
```