Screen Snapshot Service

Kurt Biery

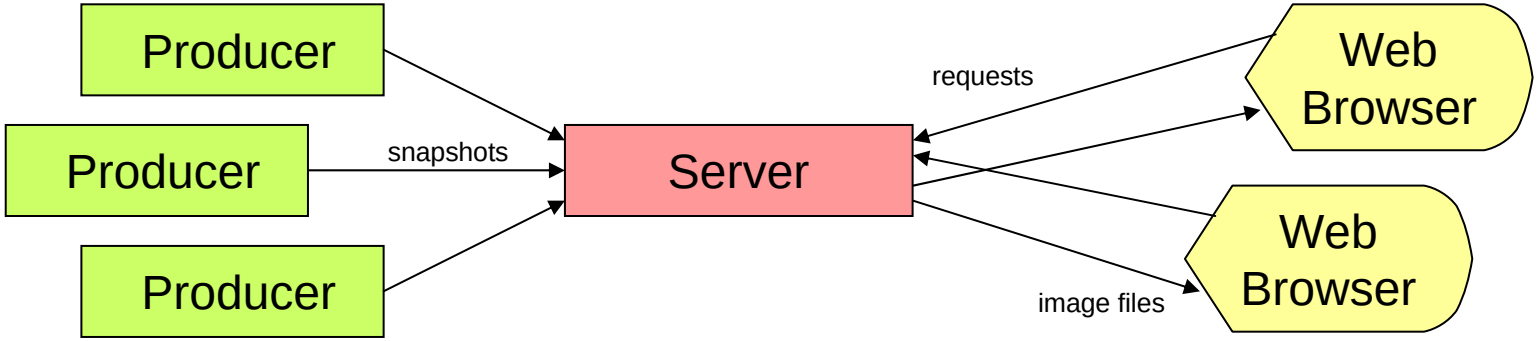
SiTracker Monitoring Meeting, 23-Jan-2007

SSS Introduction

- The goal is to provide a “snapshot” of a graphical interface to remote users.

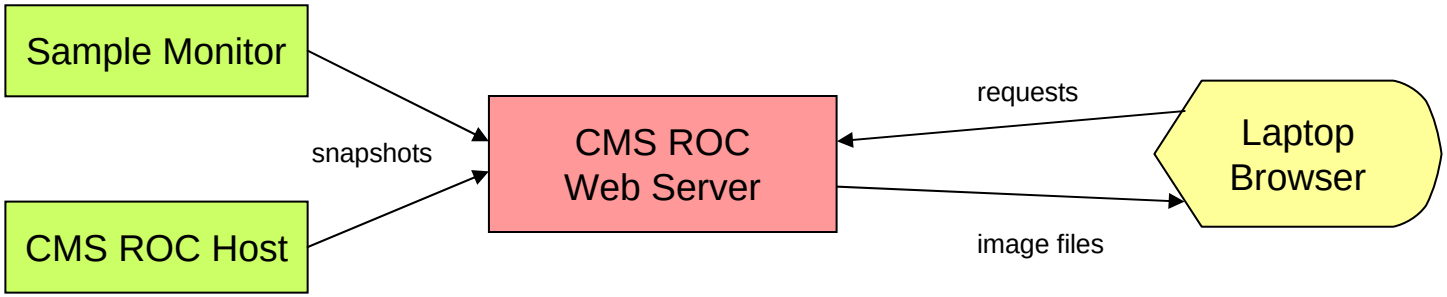
- What is a snapshot?
  - An image copy of a graphical interface at a particular instance in time.
  - Examples: DAQ system buffer display, operator control program, …
  - A view-only image, so there is no danger of accidental user input.
  - Initially implemented for desktops but could be targeted to individual application GUIs.

SSS Components



- The snapshot service is composed of three types of components:
  - producers that periodically capture snapshots of specified computer desktops and send the snapshots to a server
  - a server that receives the snapshots from the producers, converts them to image files in an easily accessible format, and serves them to clients
  - consumers that periodically fetch updated images from the server and display them to users
- In the initial version of the system, these pieces have the following implementations:
  - the producer is a Java application that runs on the host that has the display(s) of interest
  - the server is a web application that runs inside the Tomcat Java application server
  - the client is a combination of JSP pages and Javascript that execute within a web browser

SSS Demonstration



- I have some sample producers running:
  - A desktop PC in my office with a demonstration monitor GUI
  - A four-display node in the CMS remote operations center (cmsroc9)
- The CMS ROC folks have allowed me to use one of their web servers for demonstrating SSS (nippon.fnal.gov)
- We can use the following URL in a web browser to see the list of available snapshots:
  - http://nippon.fnal.gov:8080/snapshot/ShowImageList.jsp
- There is also a sample server admin page for keeping the CPU usage of the server in a reasonable range:
  - http://nippon.fnal.gov:8080/snapshot/ServerDiag.jsp

SSS Work in Progress

- Better handling of multiple displays
- User controls for the producer application
  - Dynamically change the snapshot frequency
  - Pause/resume snapshots
  - Specify number of displays and how they should be handled
- Better deployment of the producer application
- Refinements to how CPU usage is controlled
- Security controls, if needed