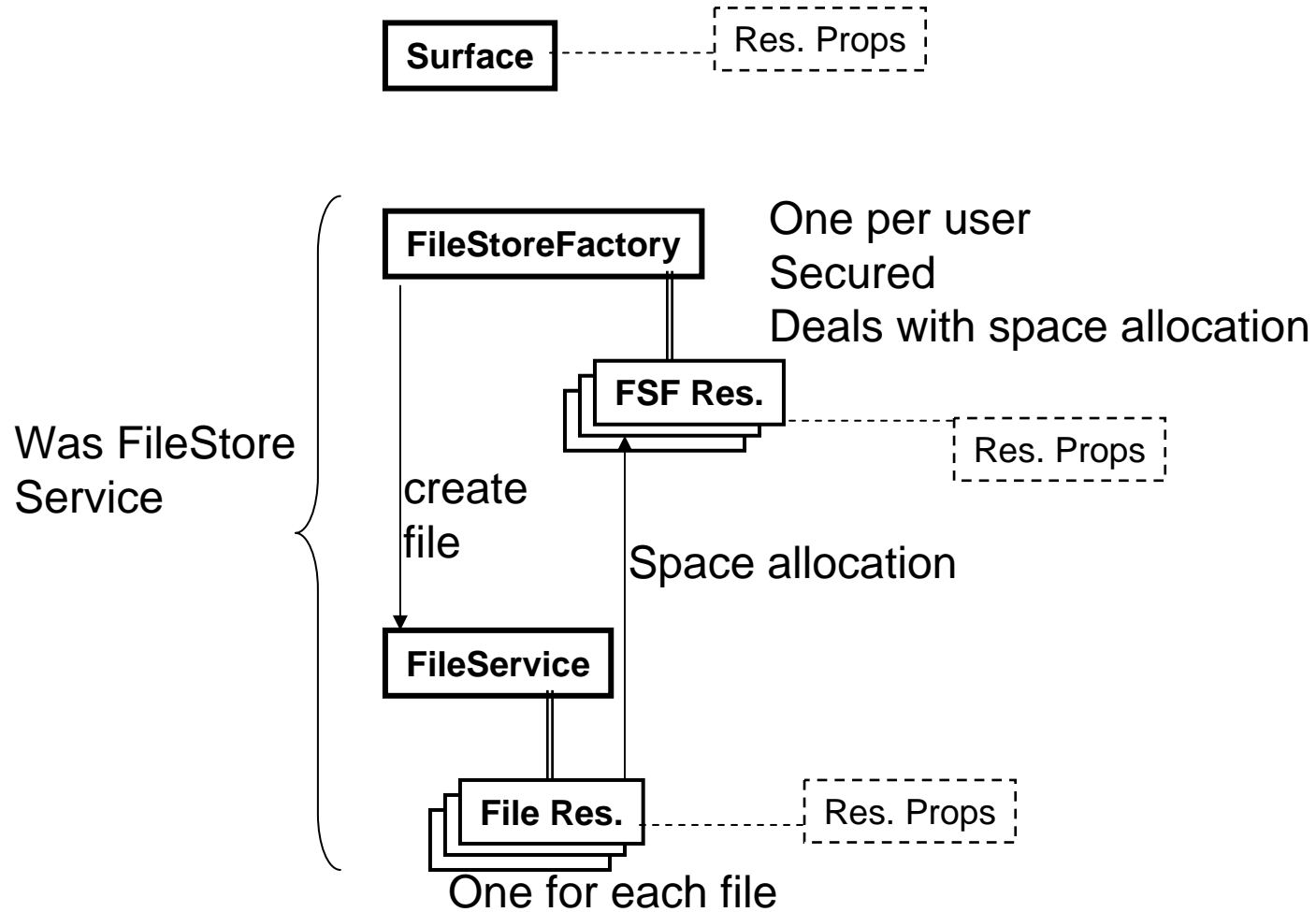
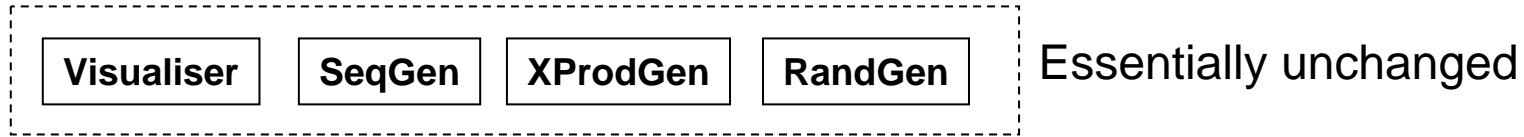


# Progressive Exercise – Using GT4 / WSRF

## Goals

- **Apply the days learning to a more substantial example**
  - **GT4 tool kit**
  - **Stateful services**
  - **WSRF mechanisms**
- **Further develop the progressive exercise with new aspects**
  - **Resource limitations**
  - **Security**
- **Preparation for Saturday's material**

# New Service Structures



# Additional Features

## Resource Identifiers

- **At all the Service interfaces –**
  - A file resource is identified by its unique EPR, allocated by File Service
- **At the CLI you need to identify some files by a meaningful name**
  - Use a file on your own machine to store the epr of the target file resource

## Resource Properties

- **Some of the services are resource homes**
  - Surface
  - FileStoreFactory
  - File
- **These have read-only resource properties**
  - Read the WSDL to discover what they are

## Lifetime – scheduled and immediate termination

- **Lifetime and scheduled termination is included**
- **Not a significant part of the exercise**

# Additional Features

## Space Allocations

- **System Resources are limited**
- **Jobs can require large resources**
- **Illustrated by file space**
- **Each user has an allocation – maximum total file size**
- **On storing data, can fail due to insufficient remaining allocation**
  - **Could be throwing away the cycles and elapsed time taken to do the job**
- **Therefore allow reservation of space on create**
  - **But create could fail unnecessarily**
- **Not much of an issue yet**
- **But later on may become important**

# Additional Features

## Security

- **Need to stop competing teams from**
  - **Stealing your resources - file space allocation**
  - **Stealing your results**

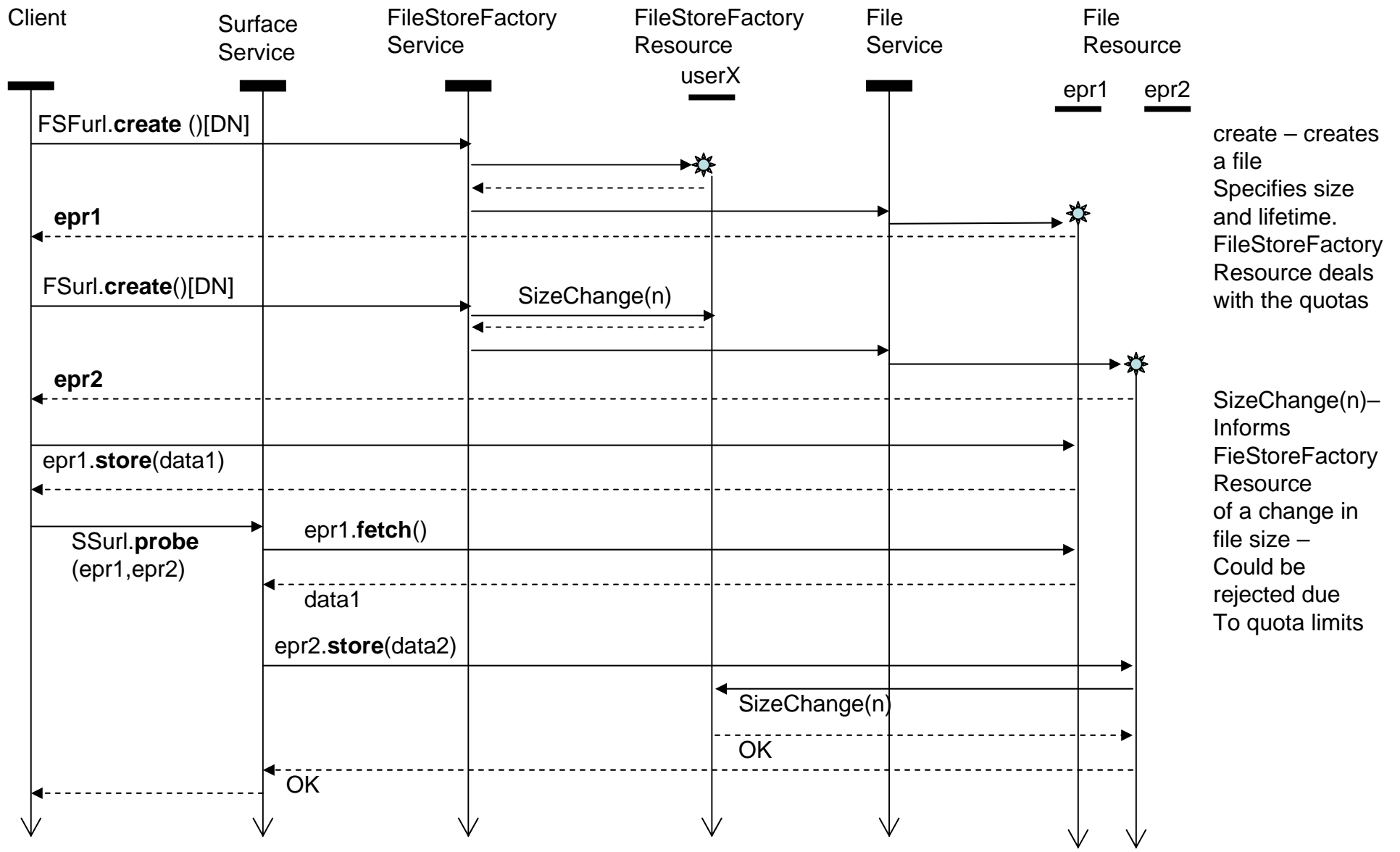
- **Security Mechanisms**

- Allocation, responsibility of FileStoreFactory Service**

- **User is identified by Distinguished Name in Certificate**
    - **FileStoreFactory Resource**
      - **Created with allocation for that user**
      - **Checked and updated whenever used filespace changes**

- File Contents, responsibility of File Service**

- **Single File Service for all users**
    - **Files are identified by EPR – an obscure identifier**
    - **If you know the file's ERP, you can read it and delete it**



create – creates a file  
 Specifies size and lifetime.  
 FileStoreFactory Resource deals with the quotas

SizeChange(n)–  
 Informs FileStoreFactory Resource of a change in file size –  
 Could be rejected due To quota limits

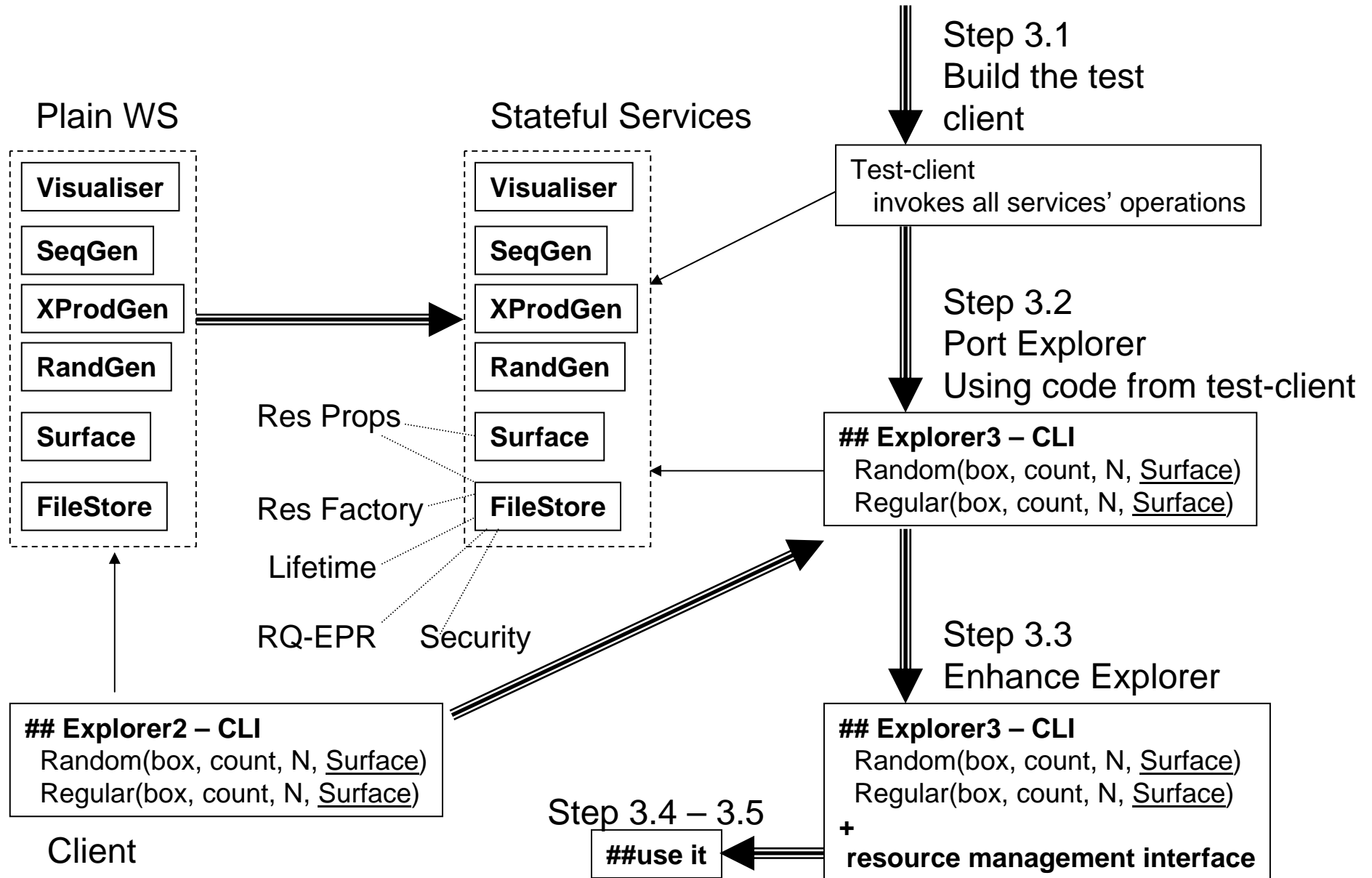
\* Creation event

XXurl.op(...) – operation, addressed to pre-known Service URL  
 eprN.op(...) – operation addressed to dynamically obtained RQ-EPR

[DN] – secured via certificate DN

In reality there are more parameters and possibly different operation names

# Structure of the Exercise



**Step 3.1 – Build the Test-Client**  
detailed instructions given

**Step 3.2 – Port your explore Client**

- **Read Test Client code**
- **Read Services WSDL**
- **Merge code from existing explorer and test-client**

**Step 3.3 – Enhance your Explorer Client**

**Steps 3.4 / 3.5 – Investigate some surfaces**

**It's all a team exercise –**  
**parts of the work can be done in parallel**



## Useful Locations

- Within `~lcc/part2/student/gridSchool0705/`

Test client source code

`ws-clients/ws/clients/src/org/globus/tutorial/tutorial/client/`

Service WSDLs

`ws-filestore/schema/filestore/`

`ws-randgen/schema/randgen`

`ws-surface/schema/randgen`

`ws-visualiser/schema/visualiser`

`ws-xprod/schema/xprod`

JavaDoc for client source

<http://www-unix.mcs.anl.gov/~ranantha/tutorial/javadocs>

- Within <http://www.gs.unina.it/repository/friday-15/>

Commented WSDLs

Exercise Instructions

- Service URLs, e.g.

- <http://localhost:8080/wsrf/services/SequenceGenerator>

- <http://localhost:8080/wsrf/services/Surface1Service>