LazySusan

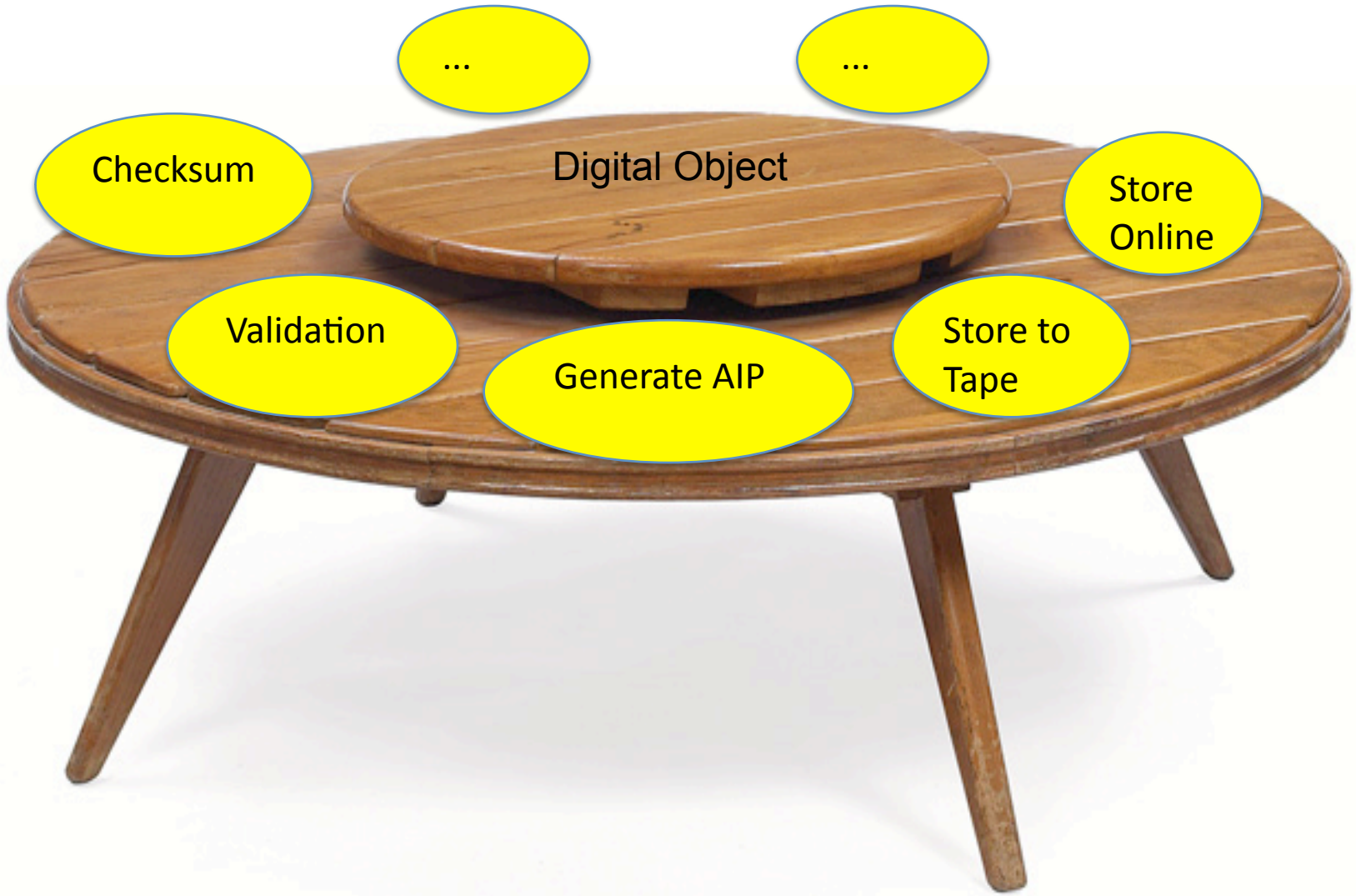
A Flexible, Scalable Digital Repository Ingest System

Alpana Pande

Stanford University

Introduction/Motivation

- Ingesting Digital Objects into the Stanford Digital Repository (SDR)
- Variable-sized DOs present challenges:
  - processing time over hours/days
  - inefficient use of tape storage
- A distributed processing model:
  - DOs are processed and stored by multiple processes on multiple machines interacting with a central job store.
- Fedora for workflow and metadata management


...
...
Checksum
Digital Object
Store Online
Validation
Generate AIP
Store to Tape

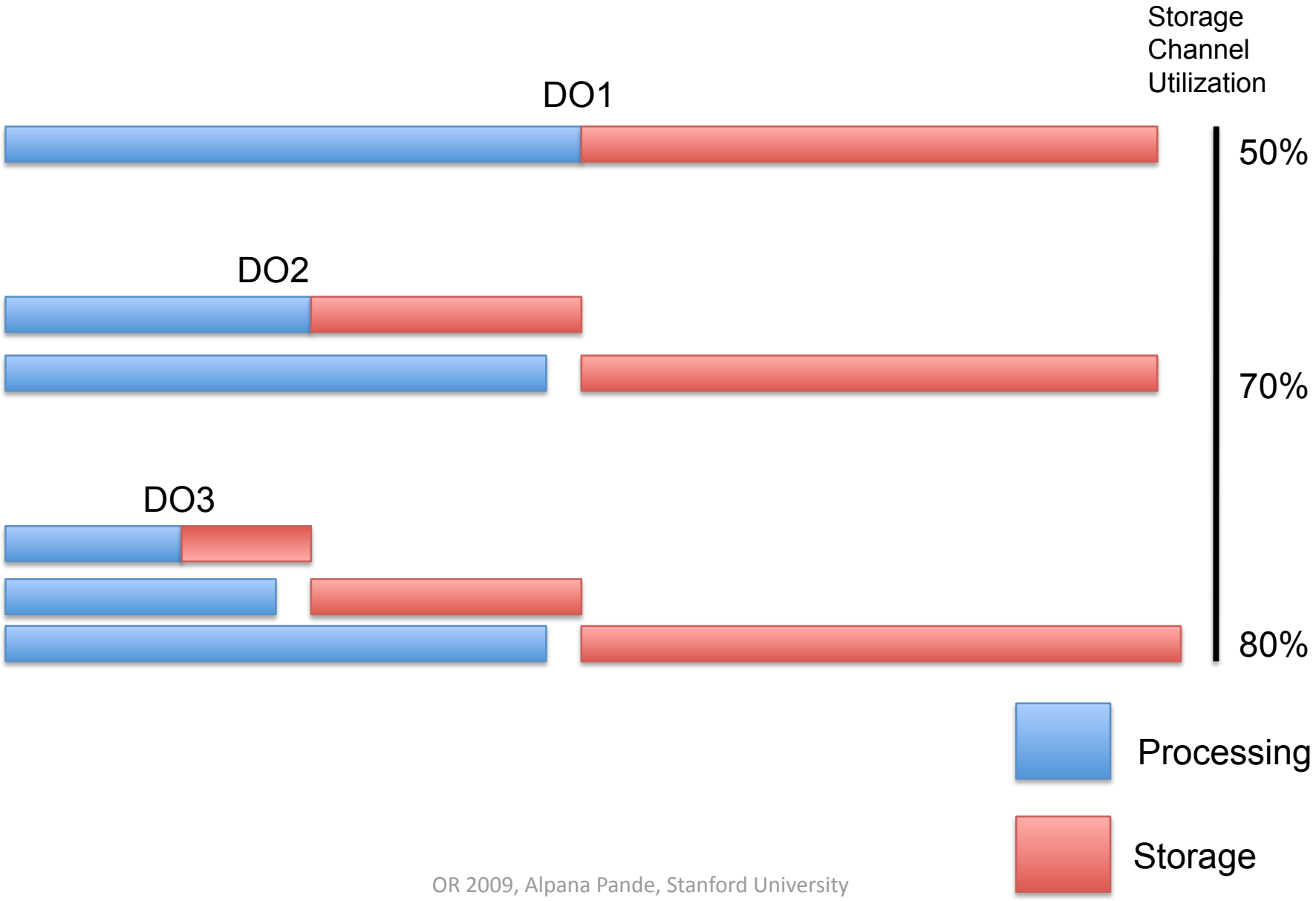
Flex/Scale

- Flexibility
  - large DOs can block smaller DOs
  - JobMaster can re-order jobs to balance large vs. small jobs, differing priorities, throughput vs. latency
- Scalability
  - processing agents (JobWorkers) are spawned and terminated on-demand to respond to changing load conditions

Storage Optimization

- Storage bandwidth is the main bottleneck in most large-scale digital repositories
- LazySusan is designed to optimize use of the storage channel
- Storage optimization scheduling arranges job processing to maximize utilization of the storage channel

Keeping the storage channel busy



Fedora for metadata management and workflow

- Every SDR DO will have a corresponding Fedora digital object created. This Fedora DO will contain administrative metadata, plus basic descriptive metadata (Dublin Core), plus content metadata
- Fedora workflow datastream to track processing status of objects through REST calls

- SDR Object
  - Header
  - AUDIT
  - DC (identifier name, value)
  - RELS-EXT
  - SDR AdminDataStream
  - SDR WorkflowDataStream
  - Content Metadata