



Production Data Export and Archiving System for the BaBar Experiment

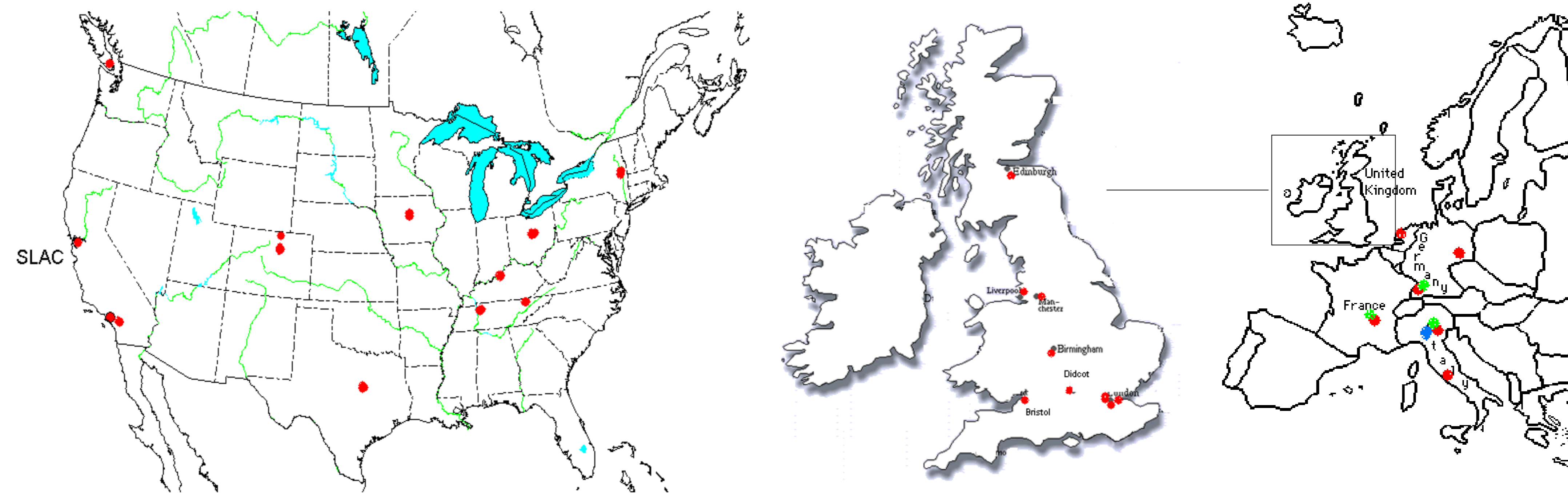
Tofigh Azemoun, Adil Hasan, Wilko Kröger, Artem Trunov
 Stanford Linear Accelerator Center
 For the BaBar Computing Group



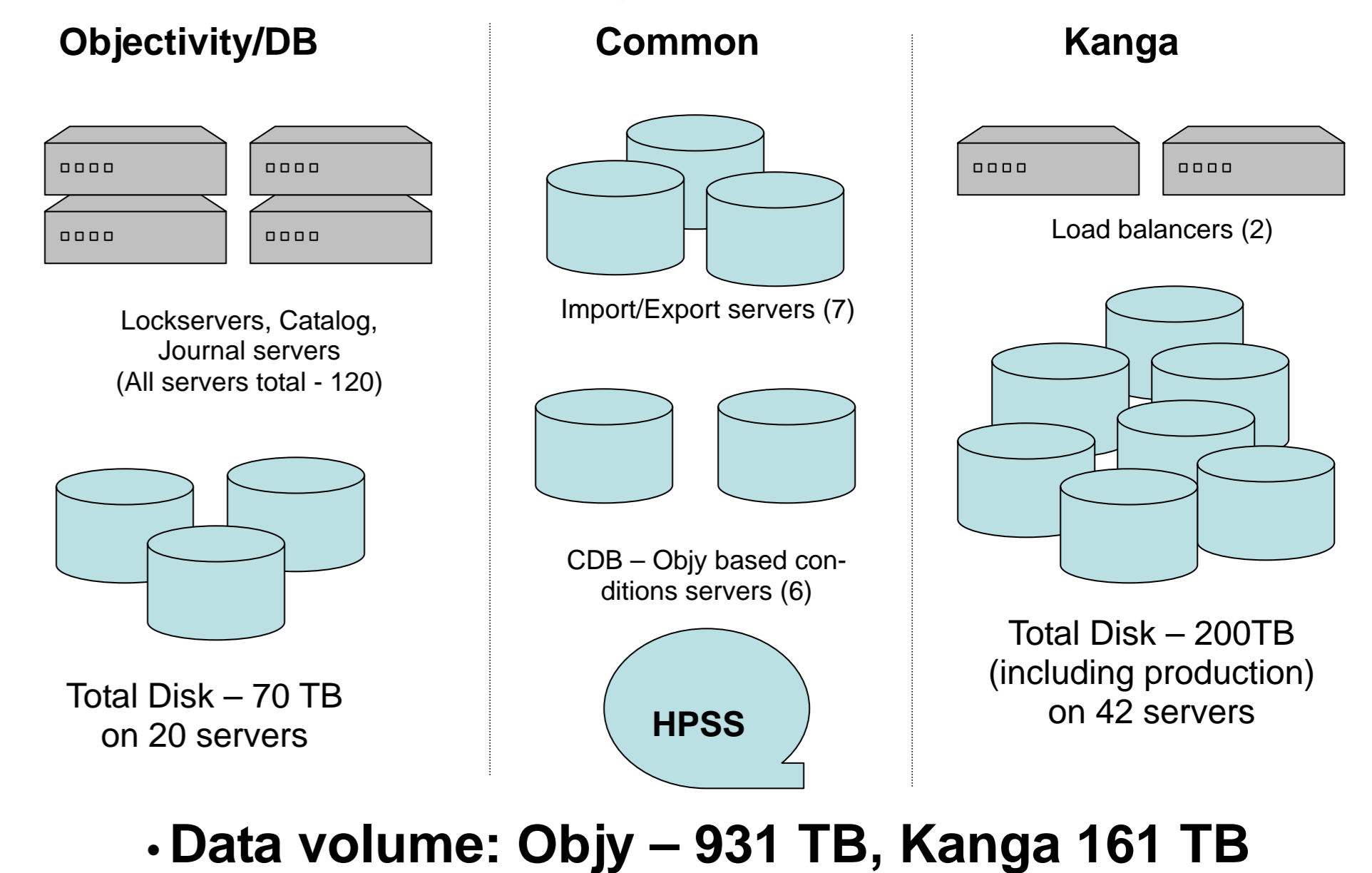
BABAR

Data Production in BaBar

- Taking data for 5 years
 - 931 TB in Objectivity/DB based (“Objy”) event store, 120 federations
 - 161 TB in ROOT based (“Kanga”) event store since Dec 2003 (including converted Objy data)
 - Distributed production
 - Event reconstruction done 100% in INFN (Italy)
 - Skimming done at In2p3, Karlsruhe, Padova
 - Monte Carlo production at about 25 sites in the US, Canada and Europe
- Monte Carlo Production sites
● Data Processing sites
● Skim Production sites



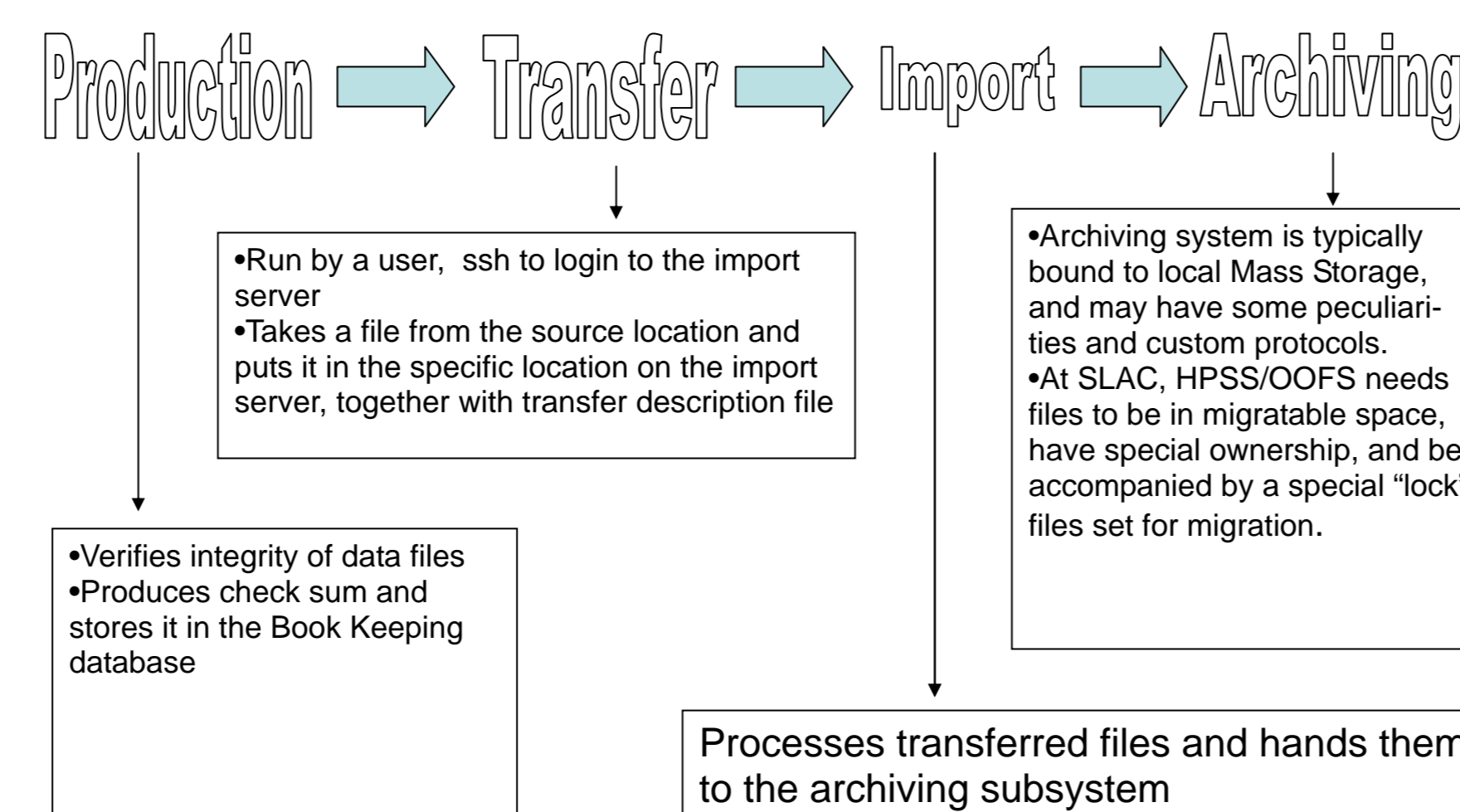
Computing Infrastructure



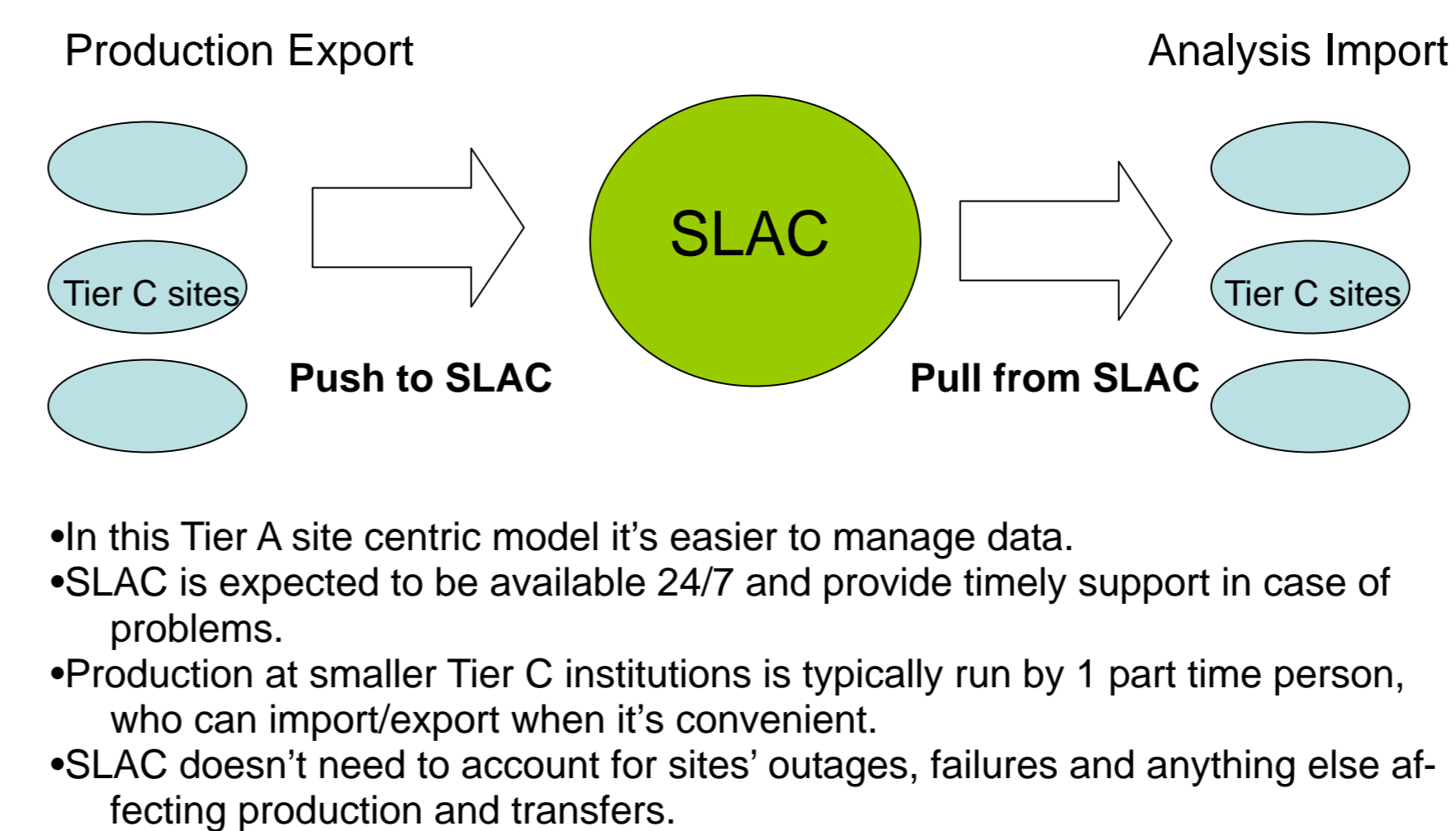
Past Experience with Data Management

- DB Administration: 2-4 people
- Different import procedures for data produced at SLAC and at remote sites
- No higher level tools to check data consistency
 - Database files could be checked with Objy utilities, but very slow
 - No tools to check collections
- Very little automation of error handling
- Can not control HPSS logic
 - Proprietary code
 - Tape mount order
 - Files write and read order
 - Congestion under heavy load from 60 servers

Sharing Responsibilities and Concerns between Subsystems



Push - Pull



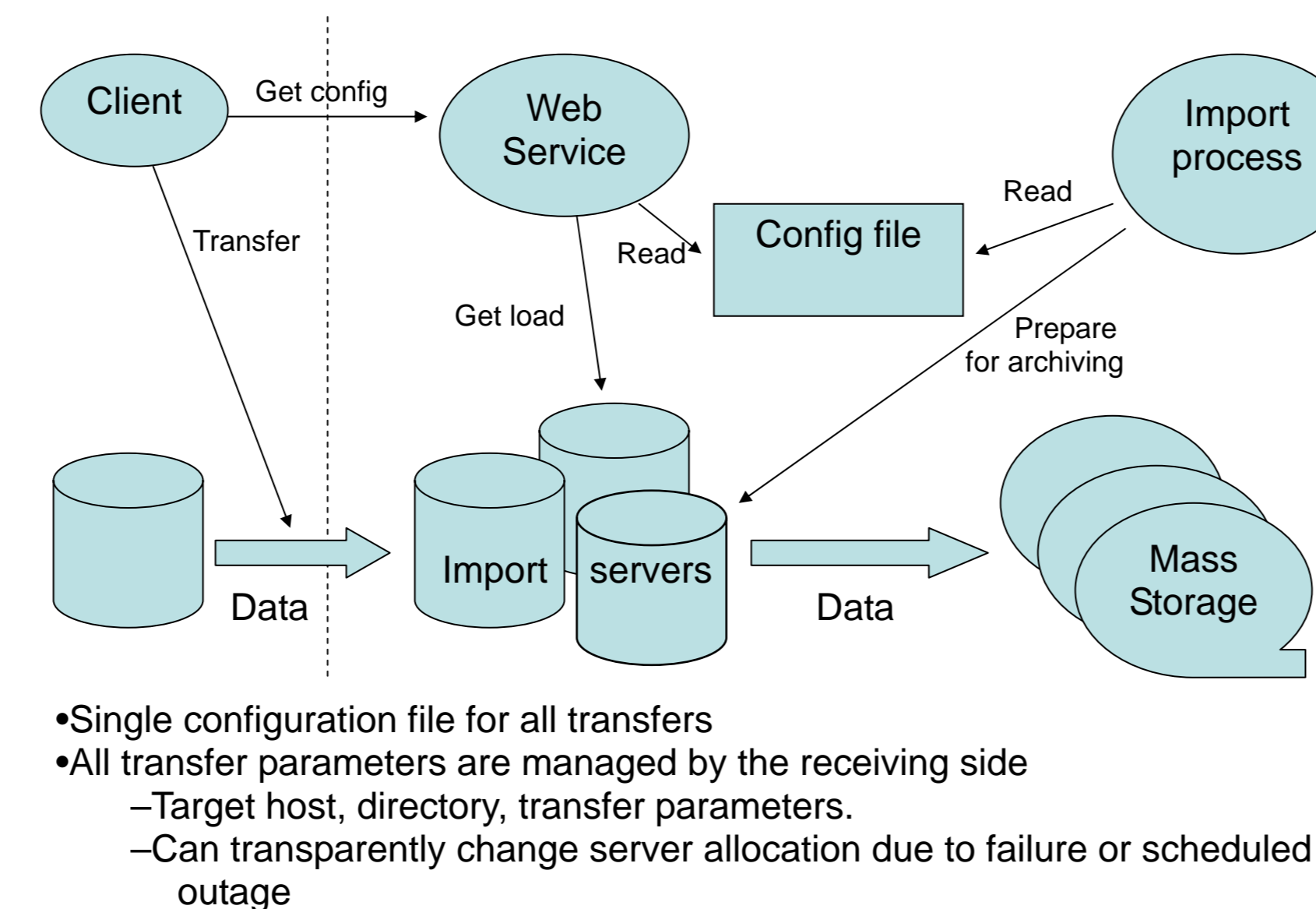
KanTransfer Tool

- Set of client and server side tools written in Perl.
- Ssh authorization for clients.
- Transferring kanga and objy collections
 - As set of files or a tar archive
- Support for custom copy programs
 - bbcp, bfbtp used in BaBar (server-less tools)
- Checksum verification
 - Compute or use supplied checksum from Book Keeping database
- Import is detached from transferring
 - Allows accumulating of data on import servers in case of problems.
- Transfer configuration as a Web service.
 - Import side decides where to place data based on the content and server load
- Easily scalable
 - Only need to add more import servers and change the configuration.

Motivation for New Development

- Full automation of data transfers and archiving
- Unify all export and archiving procedures
- Reduce human involvement in error handling
- HPSS-friendly system
- Low resource utilization, with the focus on disk bandwidth
- Protocol level backward compatibility with Objy transfer tools
- Streamlined procedures for further data processing
- Assisting with exporting data to remote institutions

Central Management



Setup at SLAC

- 5 identically configured import servers
- Each has 3 x 0.5TB filesystems with OOFS layer for HPSS access.
- OOFS automatically archives files to HPSS, purges when necessary to make room for new data.
- Only few servers write to HPSS - helps to reduce it's load.
- Import is run infrequently, allowing to accumulate data before archiving to HPSS and reduce tape mounts; and synchronously on all import servers in order to cluster relevant data on tapes.
- Import latency is 0.5-4 hours. (time between end of transfer and availability of files for users).
- Disc cache is kept 90% full at all times, making more files available online for import by external sites.
- After archiving, files are copied to read-only analysis pool, and immediately available for users.

Work in Progress

- Load-adaptive transfer
 - Adjusting network options (number of streams, TCP window) based on real transfer performance
 - Varying them and “learning” to achieve best performance
 - Taking into account “real” disk/network conditions at the time of transfer
- Failure report
 - Way for admin to debug and communicate user problems.
 - Client's configuration and errors are logged to the import server.
- Tracking status of a transfer as it passes through all stages

