dCache.org

Multi-site storage with dCache

Tigran Mkrtchyan

WLCG workshop, Manchester, 20 June 2017

Fermilab NDGF Nordic DataGrid Facility INDIGO - DataCloud Better Software for Better Science DESY HELMHOLTZ ASSOCIATION

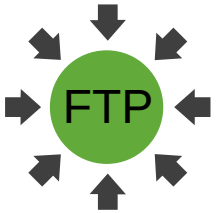
Outline or Before we go big

- How it works
  - dCache basics
- Multi-site deployment
  - Options
  - Existing installations
- Current developments

Four main components

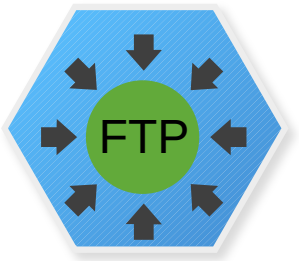
- **DOOR**
  - user entry points (NFS, FTP, DCAP, XROOT)
- **POOL**
  - data storage nodes, talk all protocols
- **Namespace**
  - metadata DB, POSIX layer
- **PoolManager**
  - request distribution unit

Minimal Setup

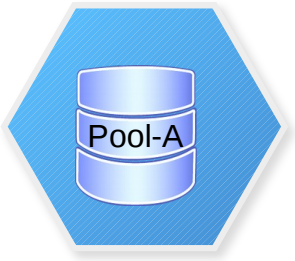


Minimal Setup

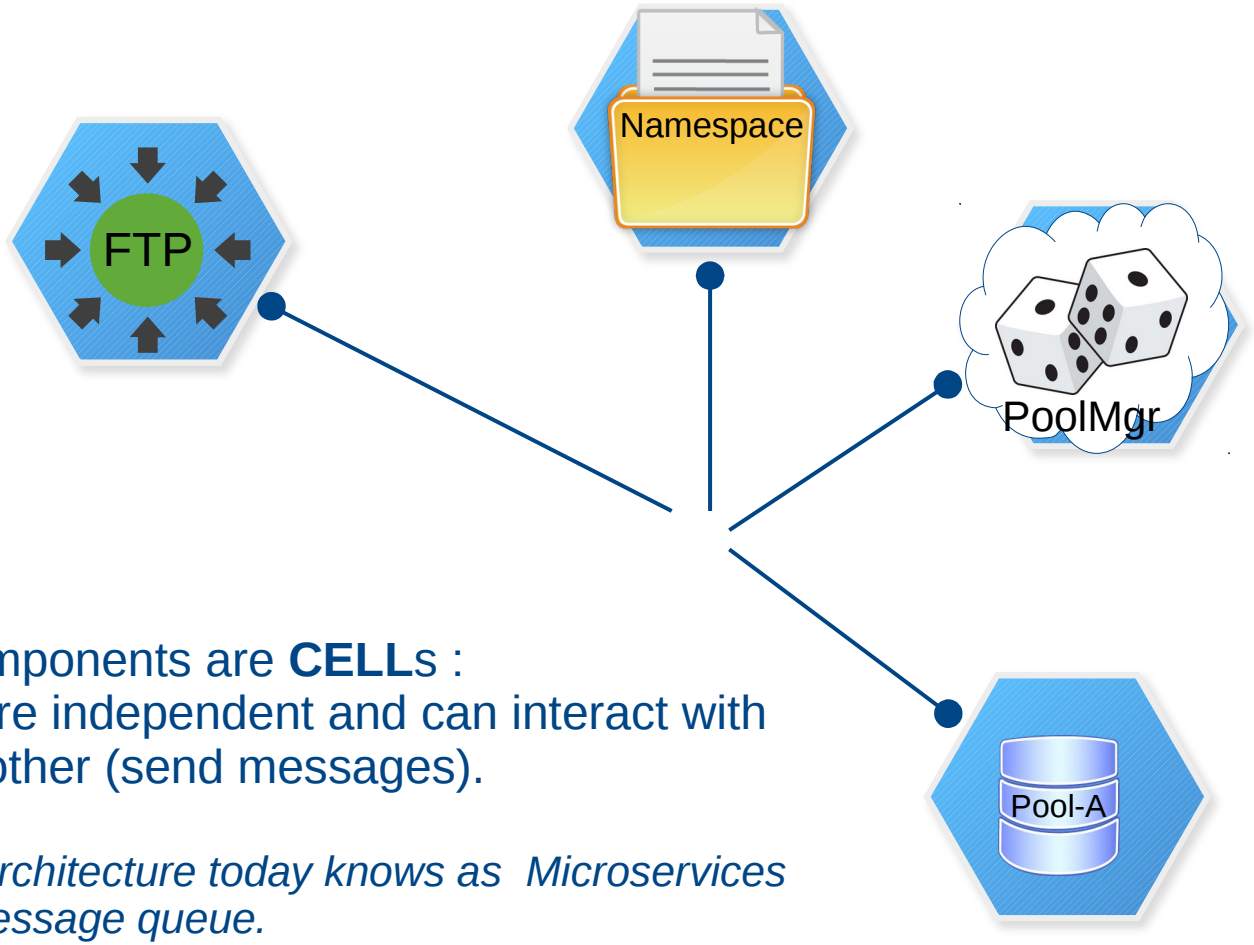
Minimal Setup



All components are **CELL**s :
they are independent and can interact with each other (send messages).



Minimal Setup

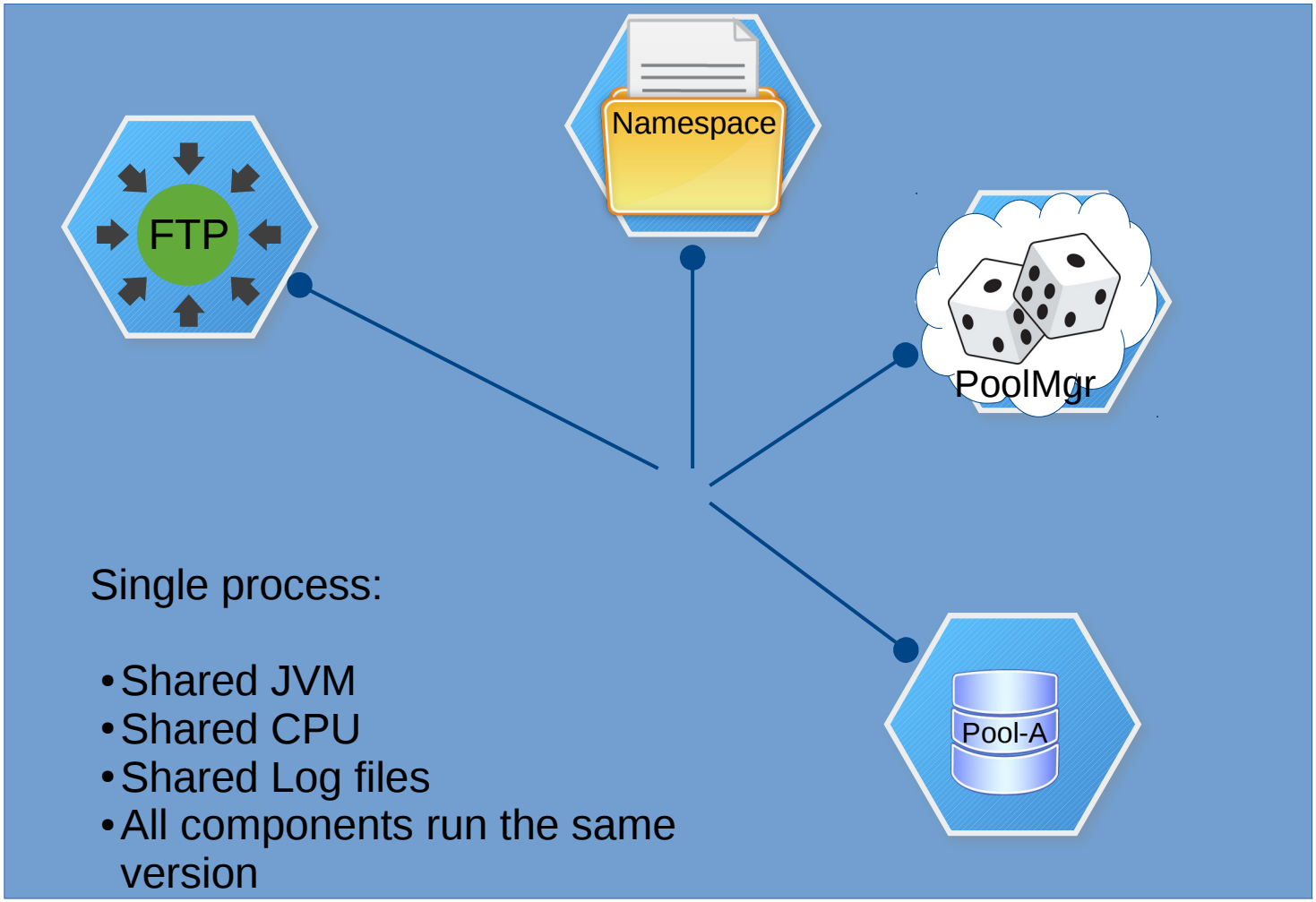


All components are **CELL**s :
they are independent and can interact with each other (send messages).

Such architecture today knows as Microservices with message queue.

Grouping CELLs

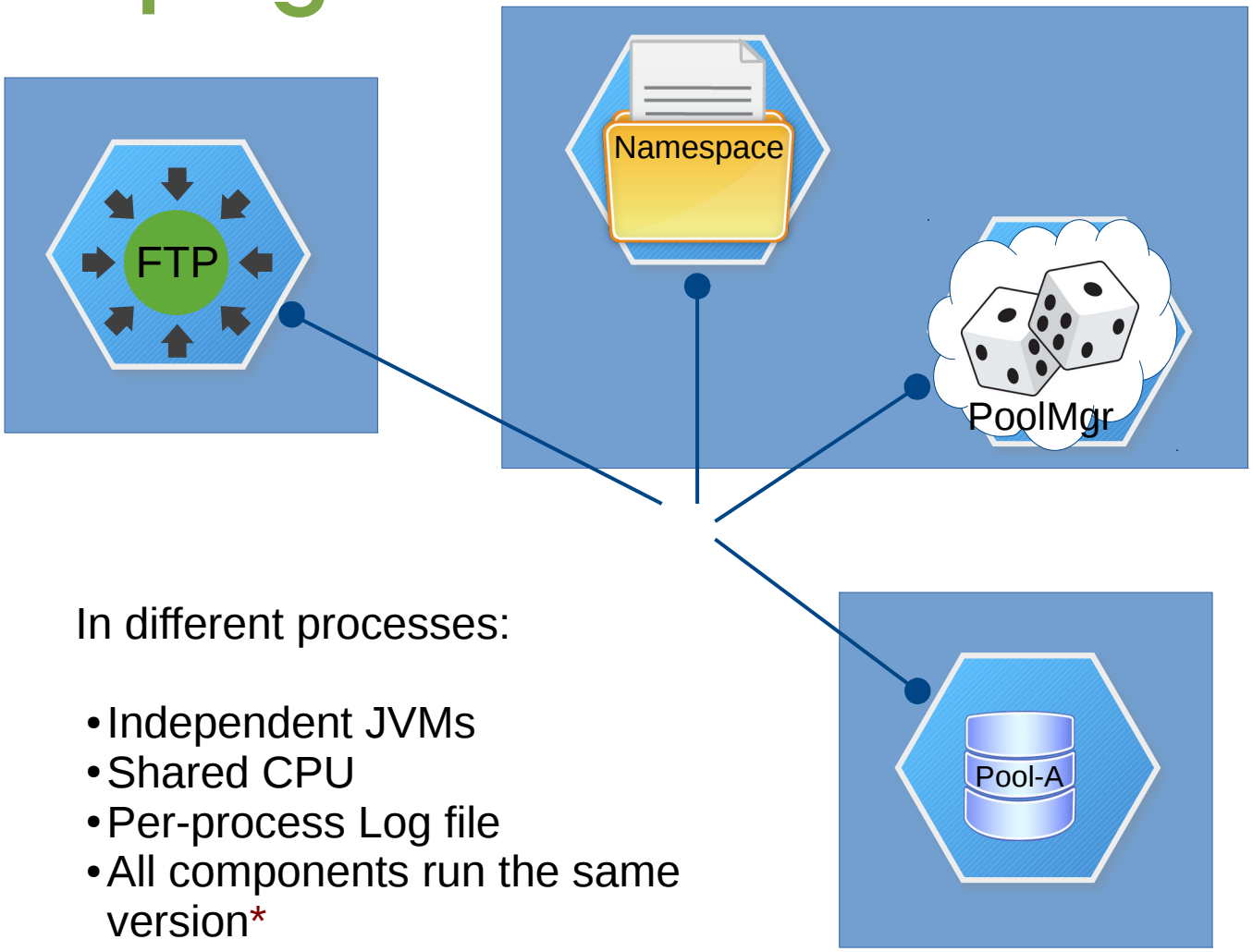
Grouping CELLs



Single process:

- Shared JVM
- Shared CPU
- Shared Log files
- All components run the same version

Grouping CELLs

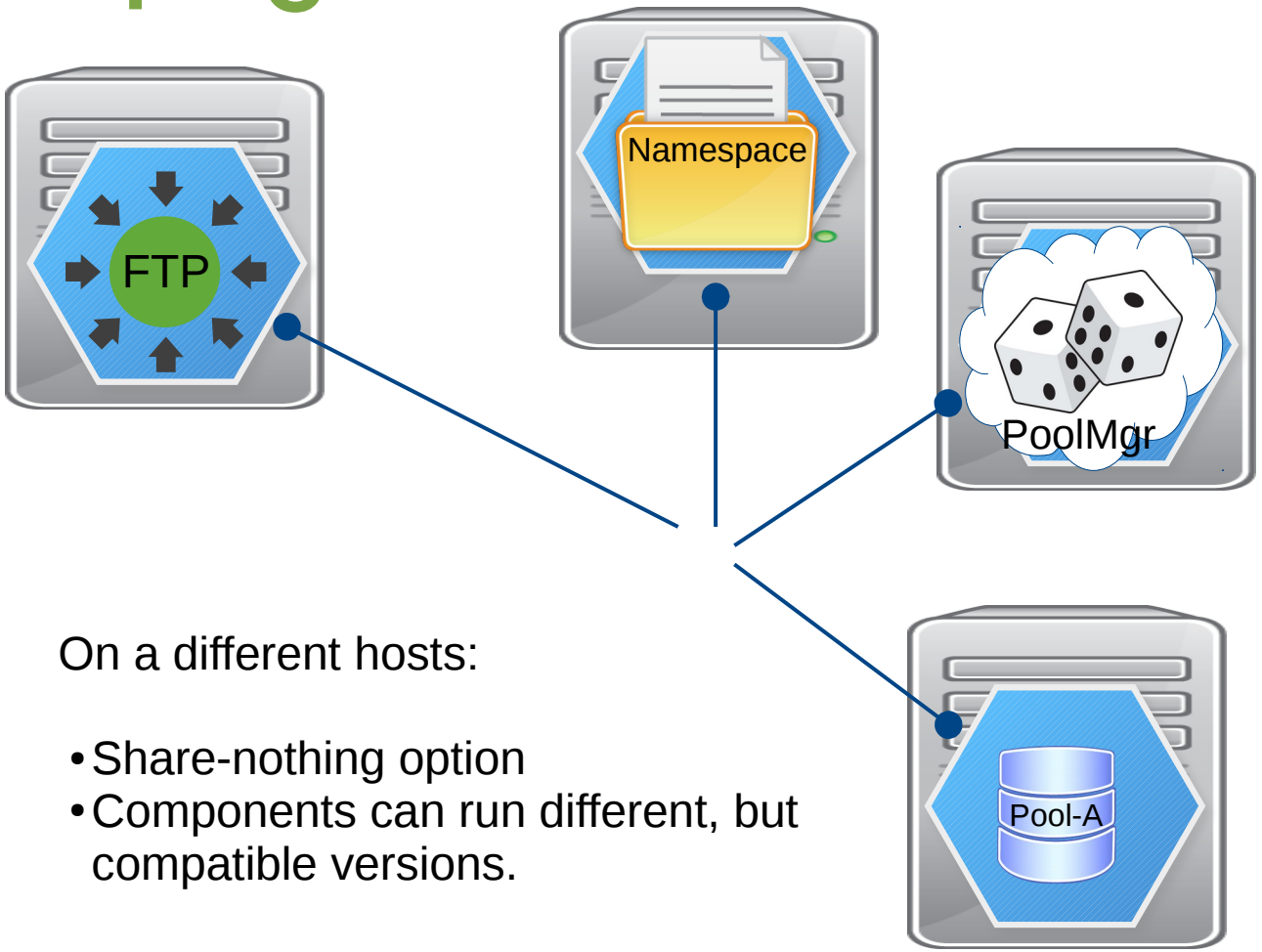


In different processes:

- Independent JVMs
- Shared CPU
- Per-process Log file
- All components run the same version*

* you can run different versions if needed

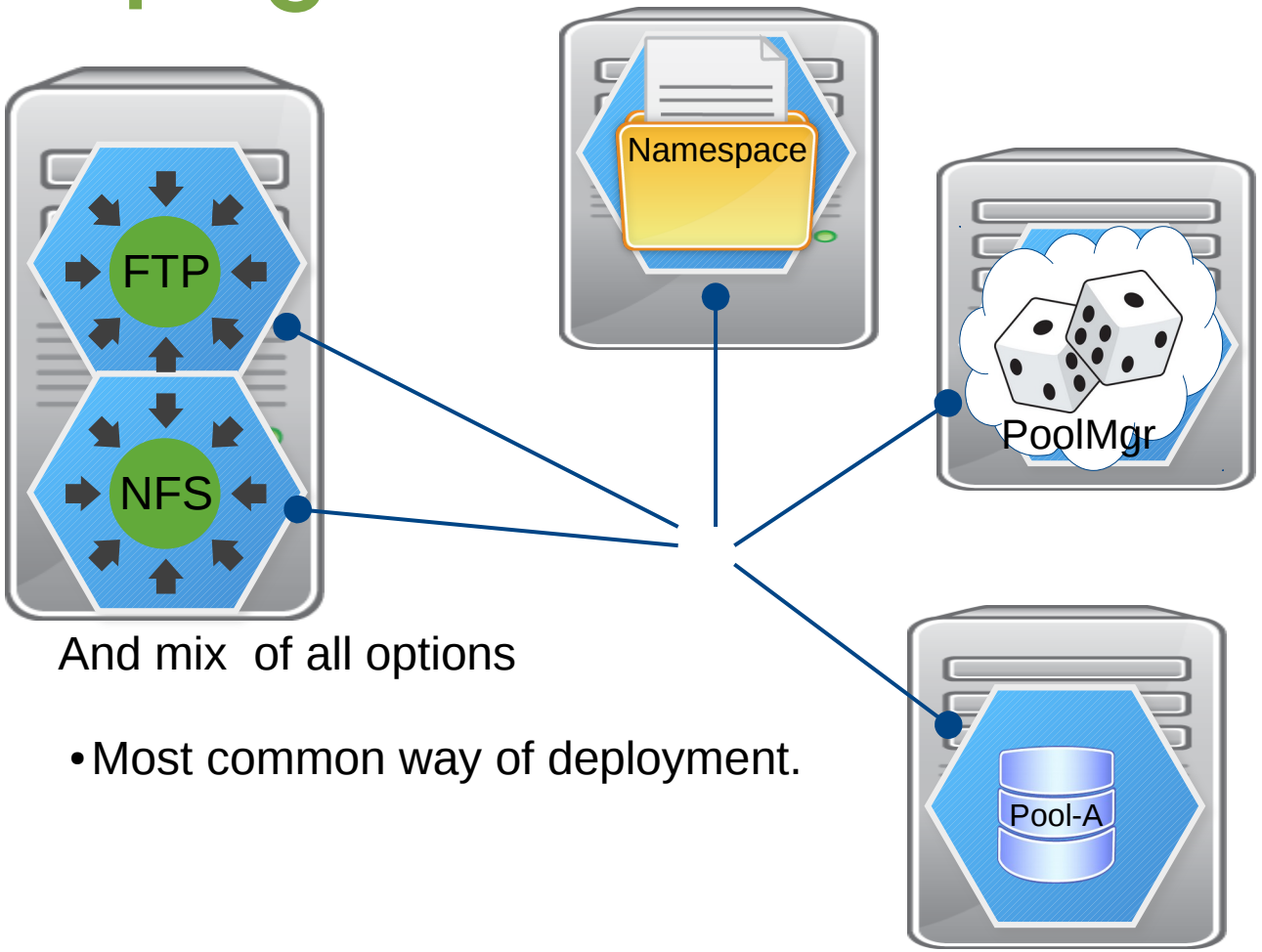
Grouping CELLs



On a different hosts:

- Share-nothing option
- Components can run different, but compatible versions.

Grouping CELLs



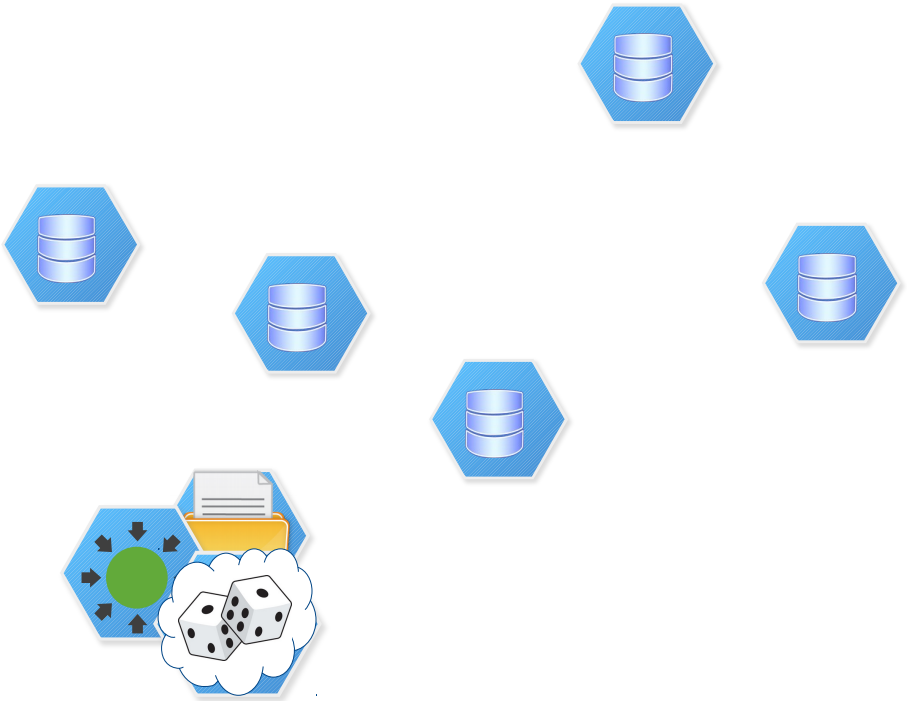
And mix of all options

- Most common way of deployment.

Multi-site deployments

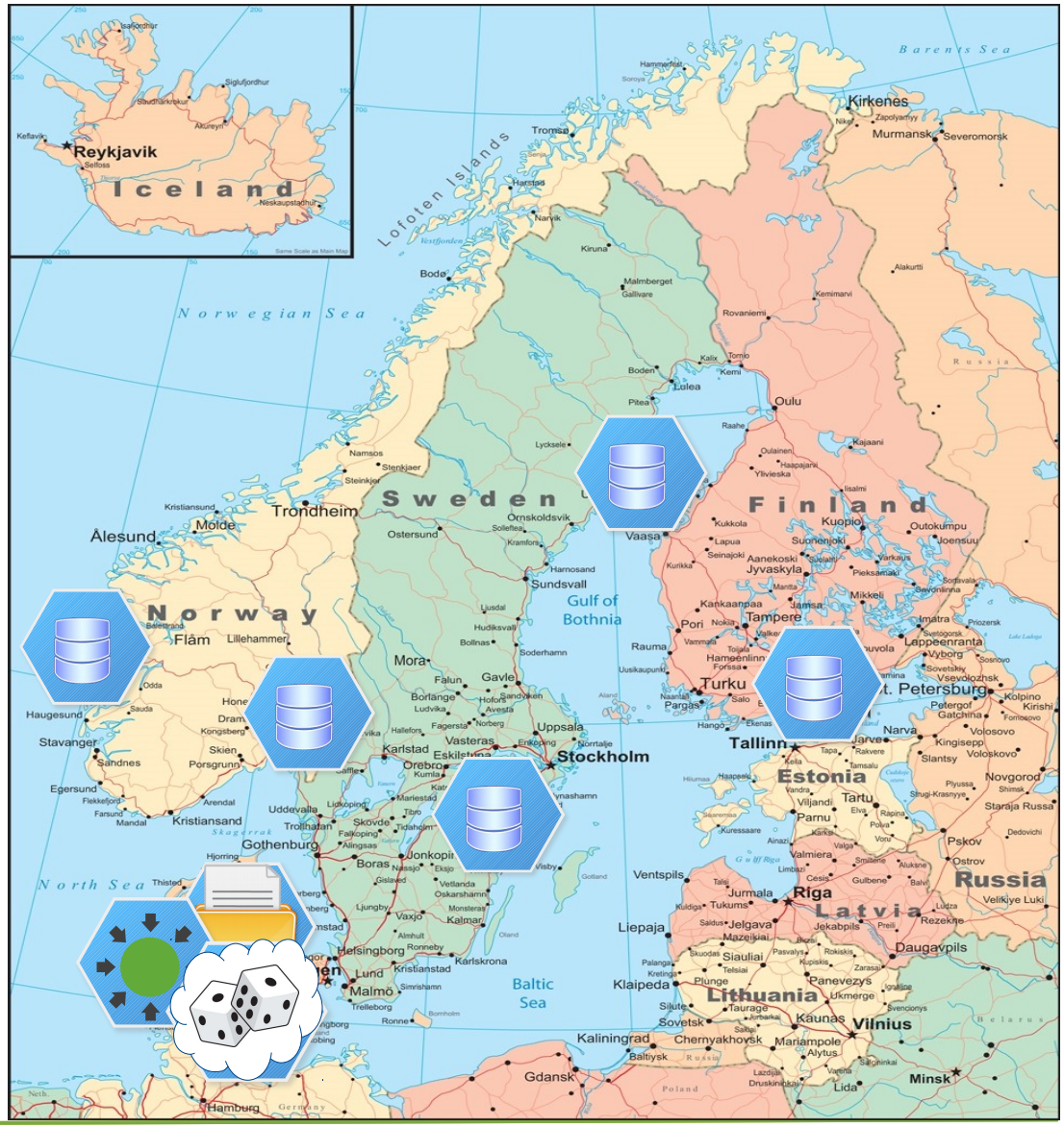
Multi-site deployment

- Distribute data over multiple locations
- Multiple administrative domains
- Use available resources



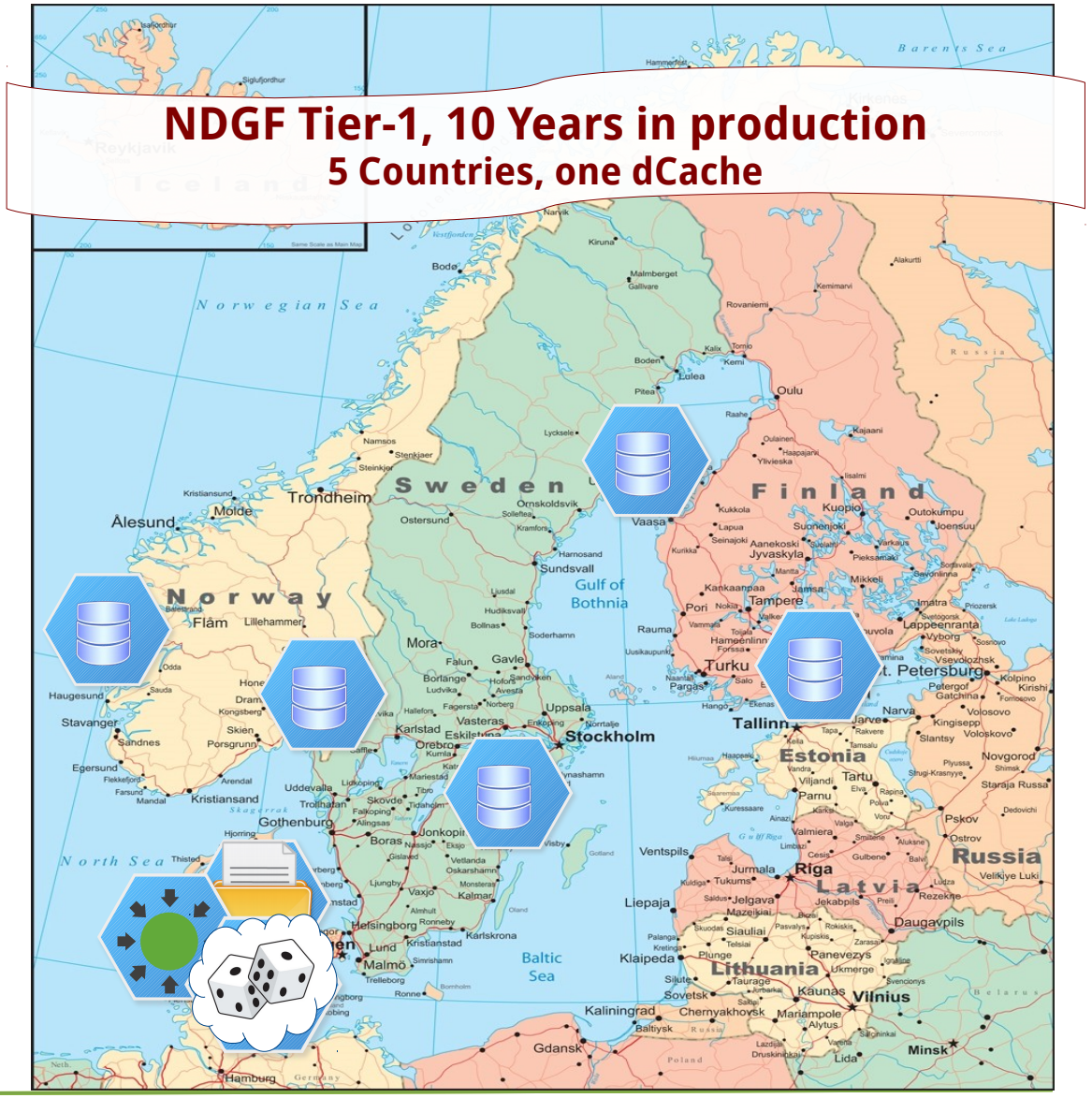
Multi-site deployment

- Distribute data over multiple locations
- Multiple administrative domains
- Use available resources



Multi-site deployment

- Distribute data over multiple locations
- Multiple administrative domains
- Use available resources

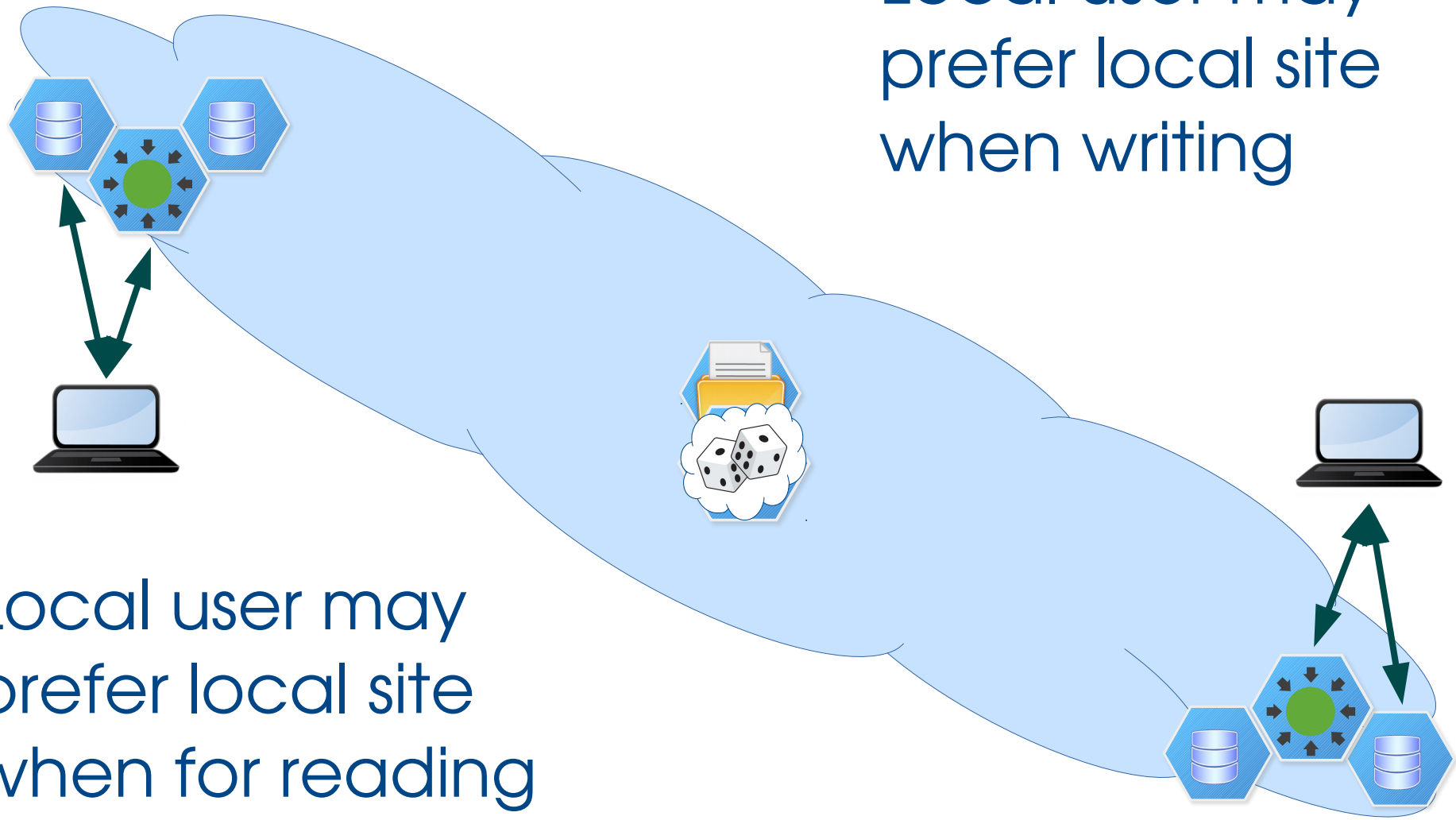


Multi-site deployment

- Works for all protocols
- Support HSM connectivity
  - Each site/pool may have it's own tape system
- Pools may run different major versions
  - Site has two years to upgrade pools

Multi-site deployment

Local user may prefer local site when writing



Local user may prefer local site when for reading

Multi-Site deployment

- Preferred write location depending on IP (location) or directory path (if requested)
- Preferred 'local' read access if data is available
- Replication
  - On Demand, when requested from remote site
  - Permanent, data protection, location adjustment
  - Manual, for data location optimization, maintenance

Multi-site deployment

Considerations

dCache.org

- Secure component communication between sites.
- Component upgrade compatibility within a major release.
- Trying the same between major releases but not always possible.
- Hot standby of headnodes possible.
- Upgrading headnode means ‘deadtime’ for the entirre system.
- “Short downtime” mechanisms are possible but never tried out.

Network traffic

- CELL messages
  - used by inter component communication
  - Starting dCache v3.2 supports TLS
- ZooKeeper
  - 3rd party product
  - used by dCache for service discovery
  - no TLS support in stable release
  - can be secured with **stunnel**

Network Authentication

- TLS based
- Client/server authentication based on host certificates
- Ideally, dedicated CA
  - Works with dCache and stunnel

Fault tolerance

- All core services can run multiple instances (replicable)
  - Namespace
  - Pool Manager
  - Space Manager
  - SRM
- Door/Pool crashes can be handled by clients
  - NFS
  - dcap
  - xrootd
- Master/slave postgres config required
  - dCache detects which node runs as master when both provided

Upgrades

- Replicable services can be upgraded at any time
- Pools/doors may require draining
- Postgres can be upgraded within same major version
  - 3rd party tools available for fail-over management

norden NordForsk neic Nordic e-Infrastructure Collaboration

HA dCache upgrades

- With the new HA features in dCache we can do system updates including reboot into new kernels with no downtime
- Can typically be done in a day, but takes a bit of watching to make sure we don't interrupt any client accesses
- Can also do dCache upgrades of headnodes without anyone noticing, over a couple of days
  - Unless something goes wrong, of course
- Hardware and headnode upgrades on different days
  - Headnode upgrades depends on haproxy draining state – this is reset by a reboot of the hardware that runs haproxy

Mattias Wadenstein, NDGF Site Report, HEPiX Spring 2017 5

Challenges

- Tracing network issues is a challenge!
- No Central OFF button.
  - No 'official' way to restart remote component
- Multiple administrative domains
  - Faulty component isolation.
  - Log files inspection.
- Scheduling of down-times.

Summary and Conclusions

- dCache has a long tradition in providing federated storage for WLCG
- The configuration flexibility allows to control data placement and replication
- Fault-tolerant setup is recommended for a distributed deployment
- We solve technical issues, sites have to coordinate federated setup operation