z/TPF V1.1

TPF Users Group - 2011
Title: *z/TPF Database Options*

Name: Mark Cooper
Venue: Database Subcommittee

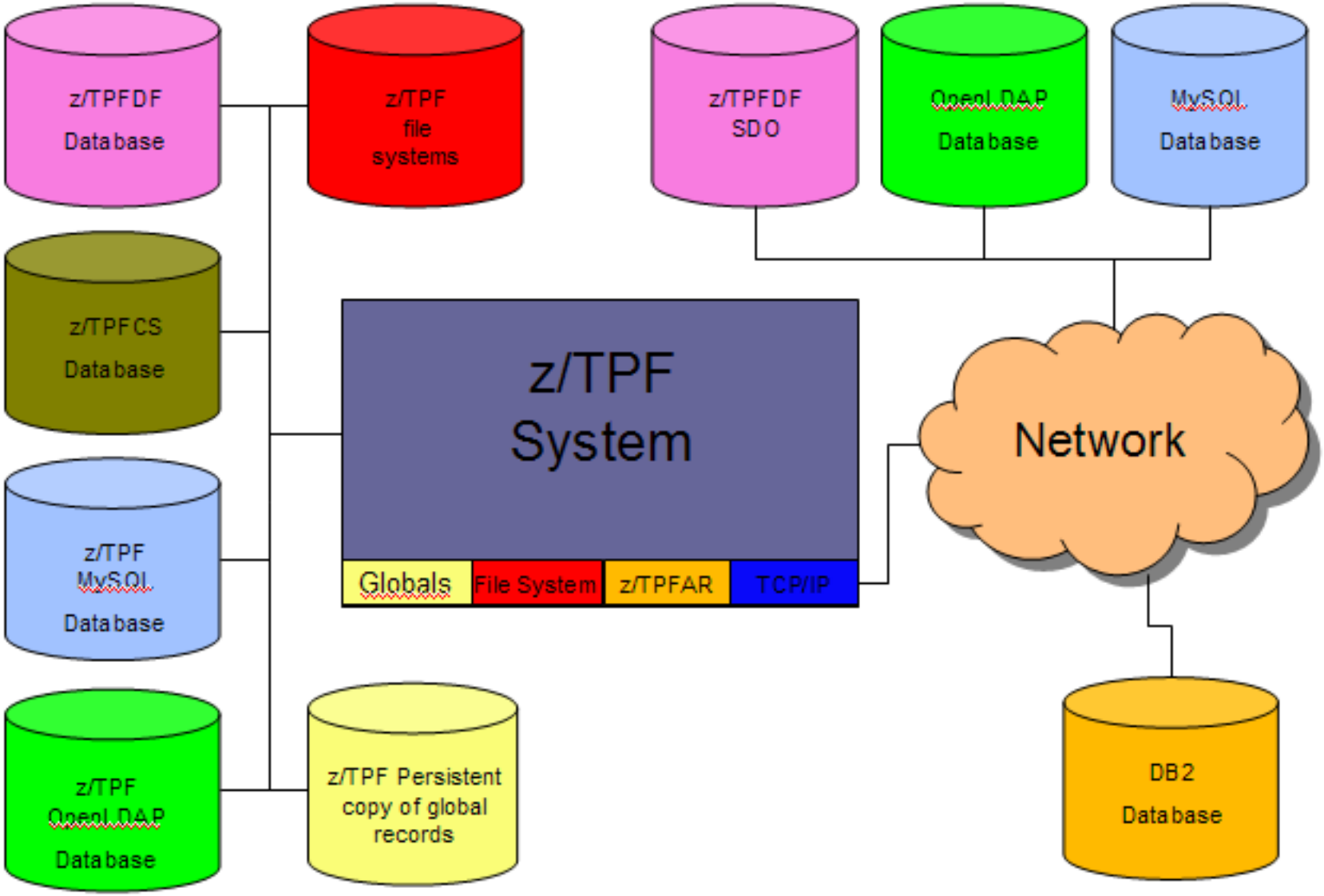
AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

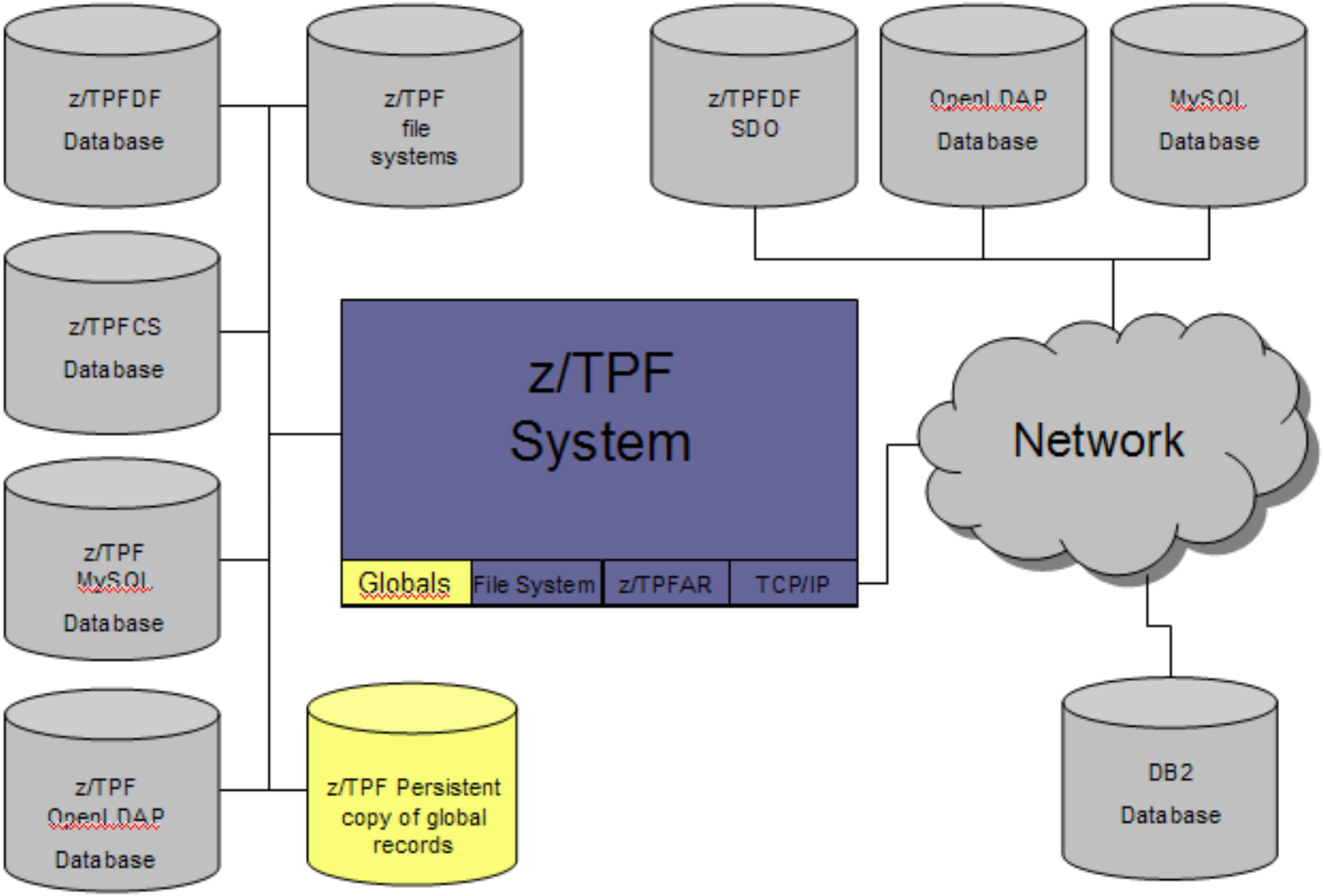
Agenda

What database options are available on z/TPF?

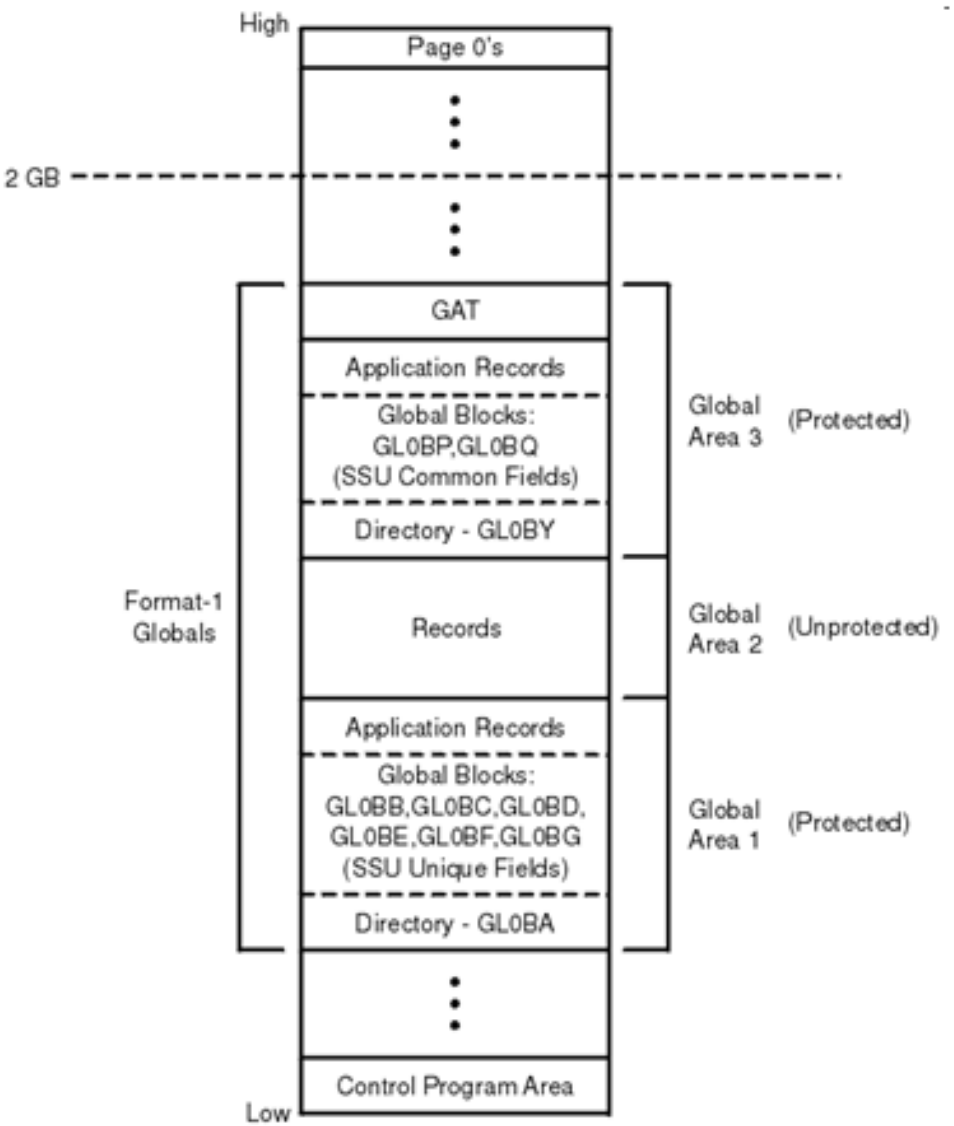
What are the advantages/disadvantages of each option?

Does IBM recommend any options?





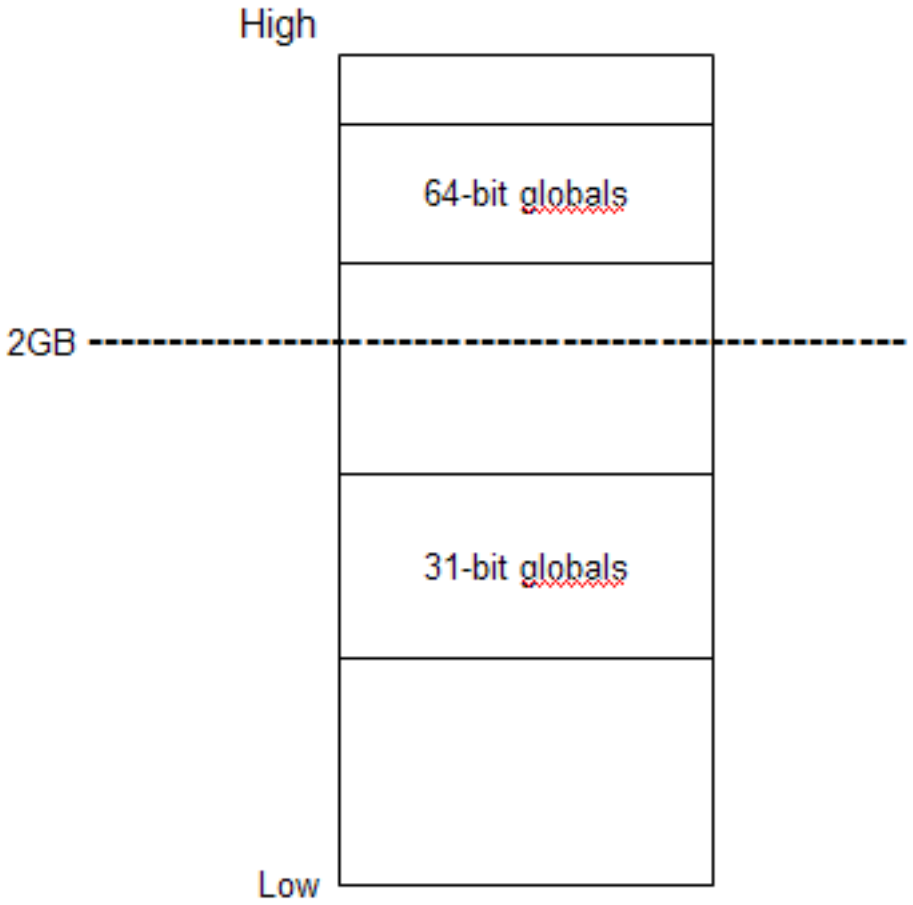
z/TPF format-1 globals support



z/TPF format-1 globals support

- **Benefits**
  - Unmatched access speed (2 instructions)
  - Great for very heavily accessed data fields and records
  - Customizable uniqueness characteristics (I-stream, processor, subsystem user)
  - Keypointable or synchronizable for persistence
- **Drawbacks**
  - TPF-unique APIs
  - Difficult to manage
  - Records are limited to 1055 bytes
  - Limited number that can be defined
  - Not easily accessible from remote platform
  - Loosely Coupled (LC) support limited

z/TPF format-2 globals support



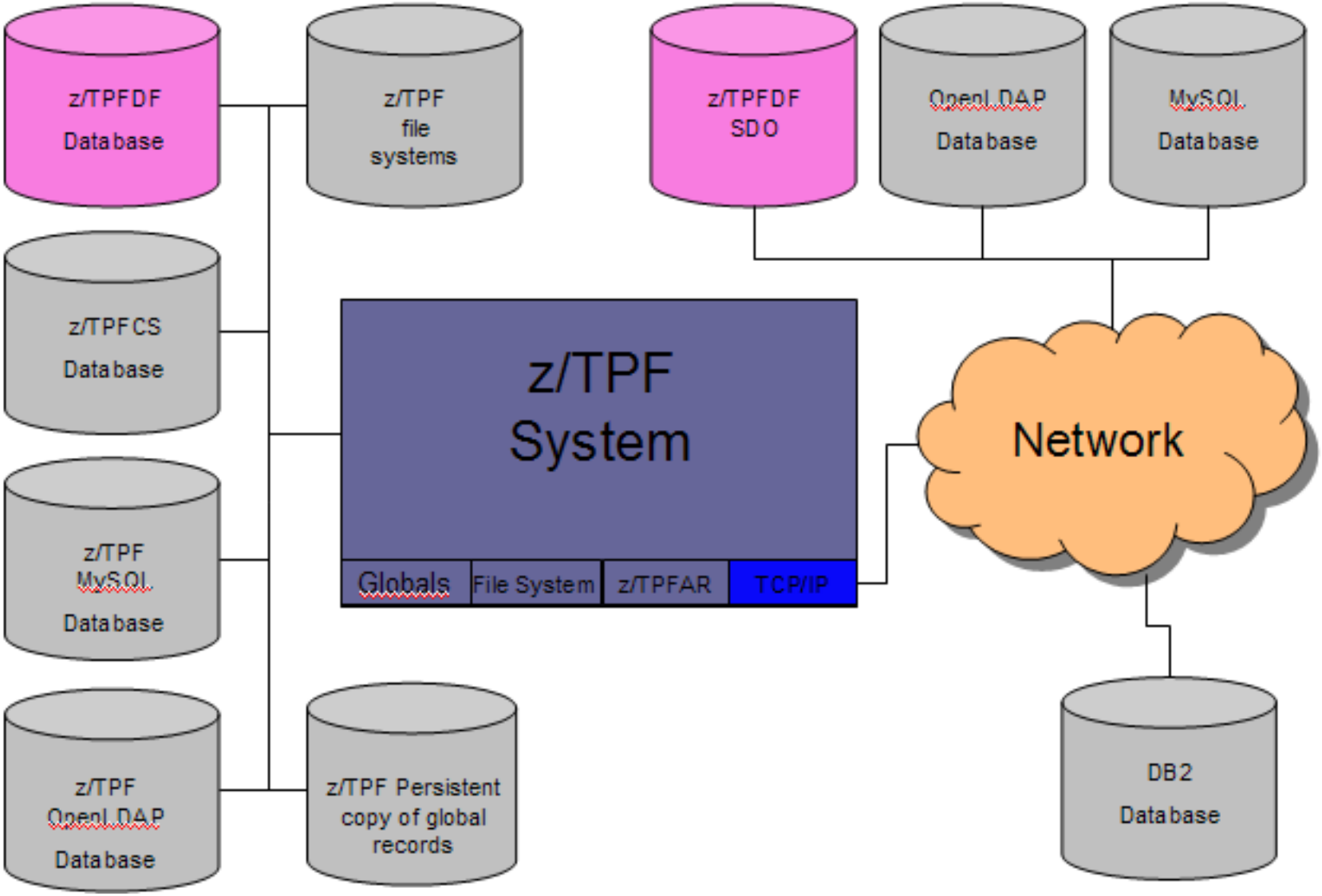
Reside either below the 2-GB bar (31-bit globals)

or above the 2-GB bar (64-bit globals),

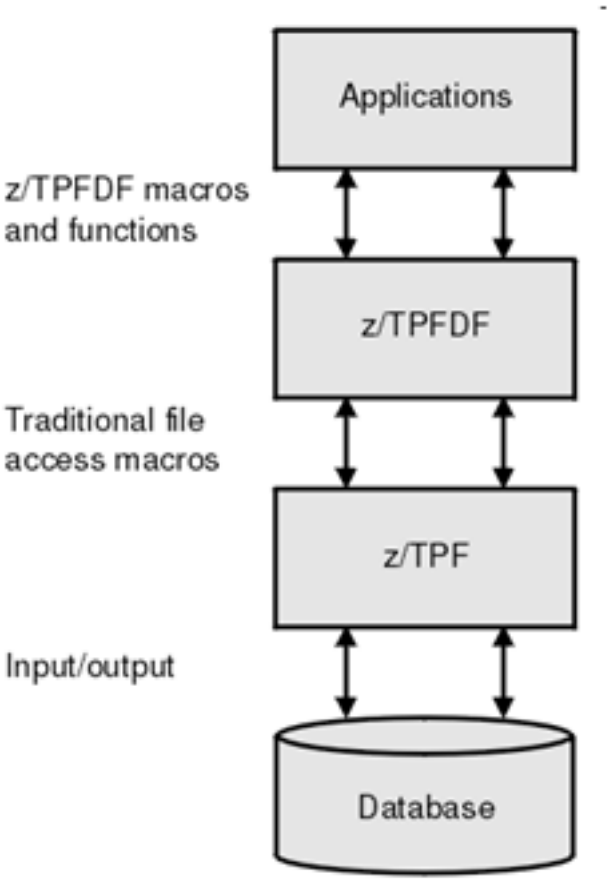
in dynamically allocated areas of system storage.

z/TPF format-2 globals support

- **Benefits**
  - Very efficient read-only access speed (20 instructions)
  - All of the benefits of format-1 globals BUT:
    - With improved manageability
    - Without the size and number restrictions
- **Drawbacks**
  - TPF-unique API
  - Not easily accessible from remote platform
  - LC support limited



z/TPF Database Facility (z/TPFDF)



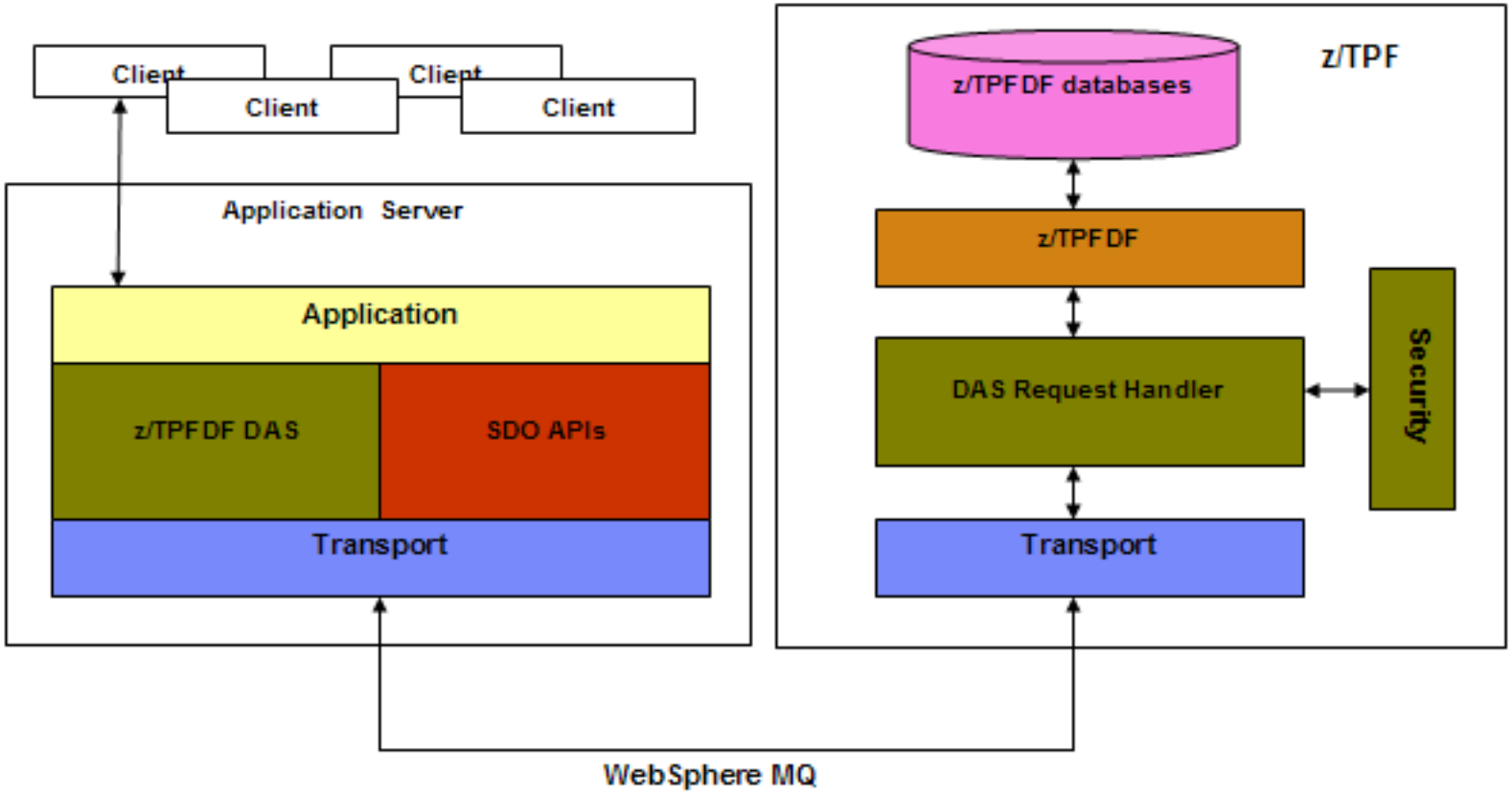
z/TPF Database Facility (z/TPFDF)

- **Benefits**
  - Generally most efficient mechanism for accessing and updating TPF data without using FIND/FILE directly.
  - Provides a level of abstraction and easy to understand database APIs and management capabilities
  - Service data object (SDO) access to data from remote Java platforms
- **Drawbacks**
  - TPF-unique API (on TPF)
  - Non-standard management utilities

z/TPFDF and Service Data Objects (SDO)

- New model of data access
- Complementary technology for SOA
- Developed jointly by IBM and BEA
- Standardized using Java Specification Request (JSR) 235
- Supported by Open SOA Collaboration
  - http://www.osoa.org/
  - IBM, BEA, Oracle, SAP, Sun, Sybase, etc

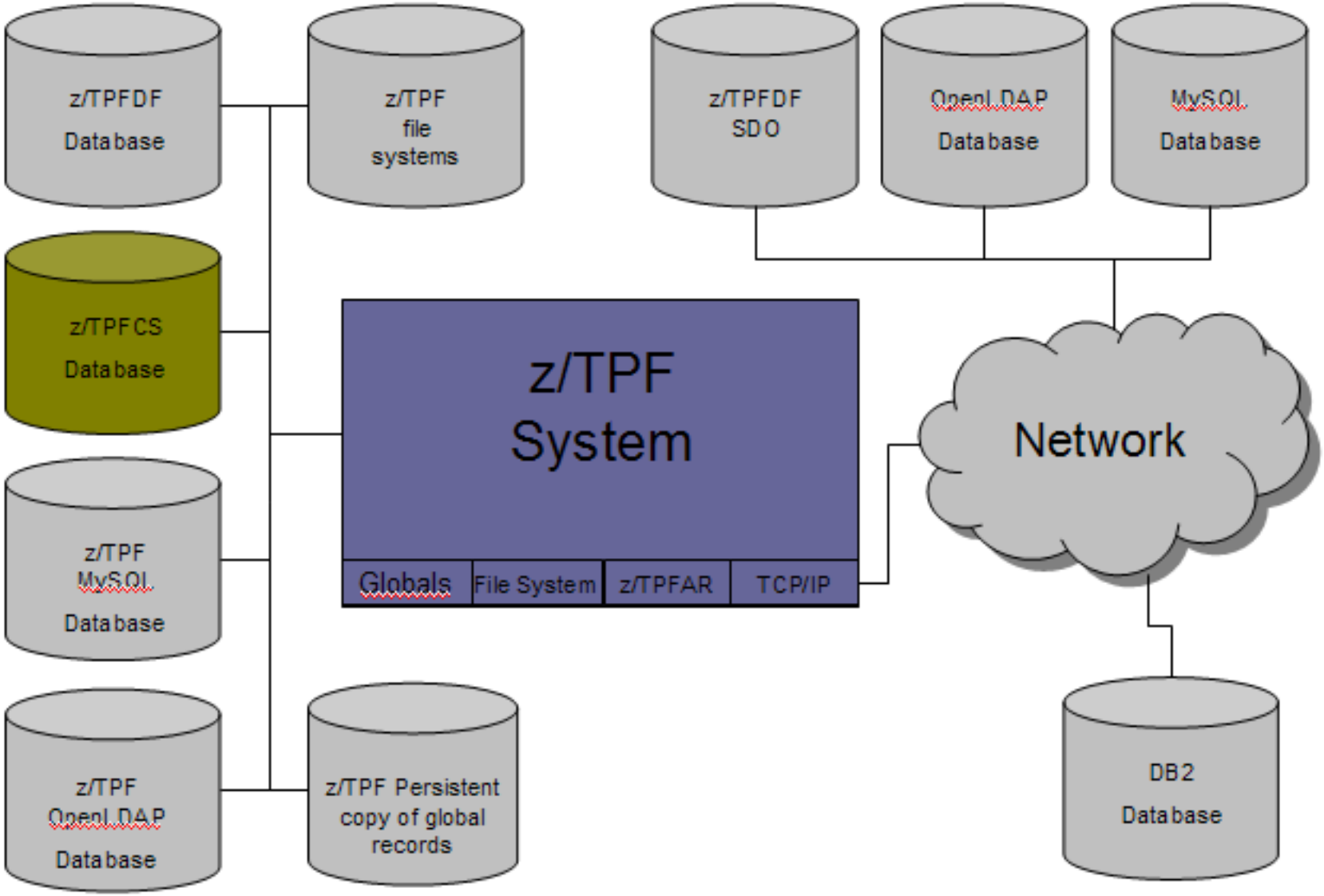
SDO: Component overview



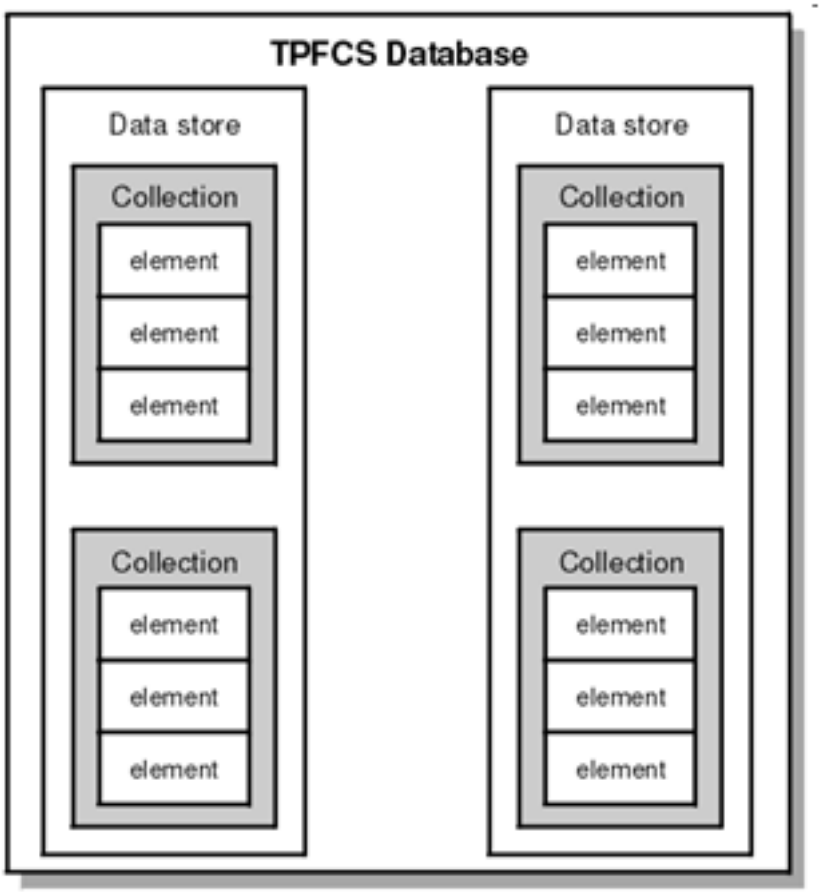
Note: Items shipped from TPF development lab are in blue, green and brown

SDO: Benefits

- Convenient and generic way to access z/TPFDF data from a remote platform
- Universal model for business data
- Common unifying format for exchanging data between services
- Includes dynamic interfaces
- Not tied to the data organization, like SQL to relational databases
- Object-oriented, thus maintenance is easier

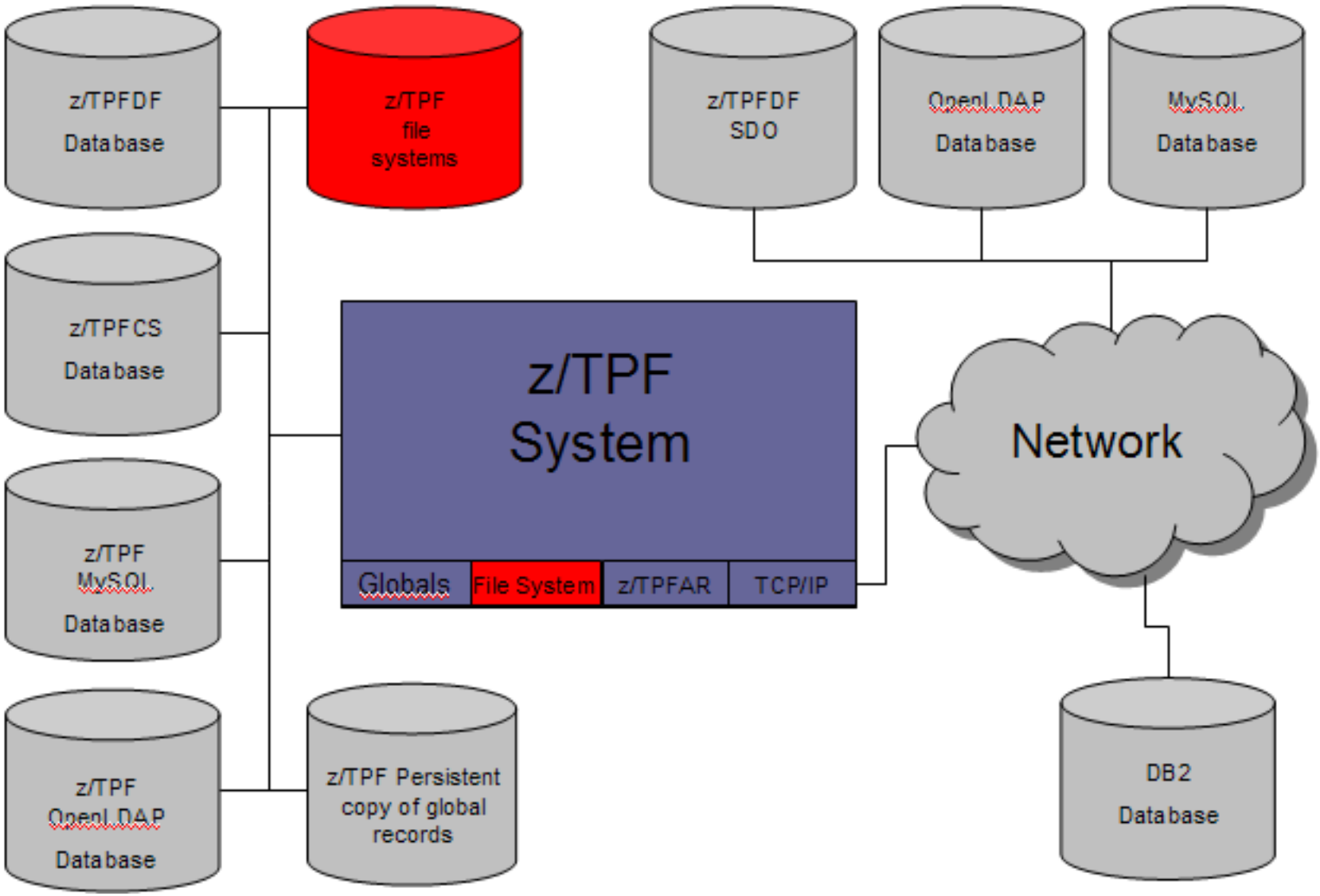


TPF collection support (TPFCS)

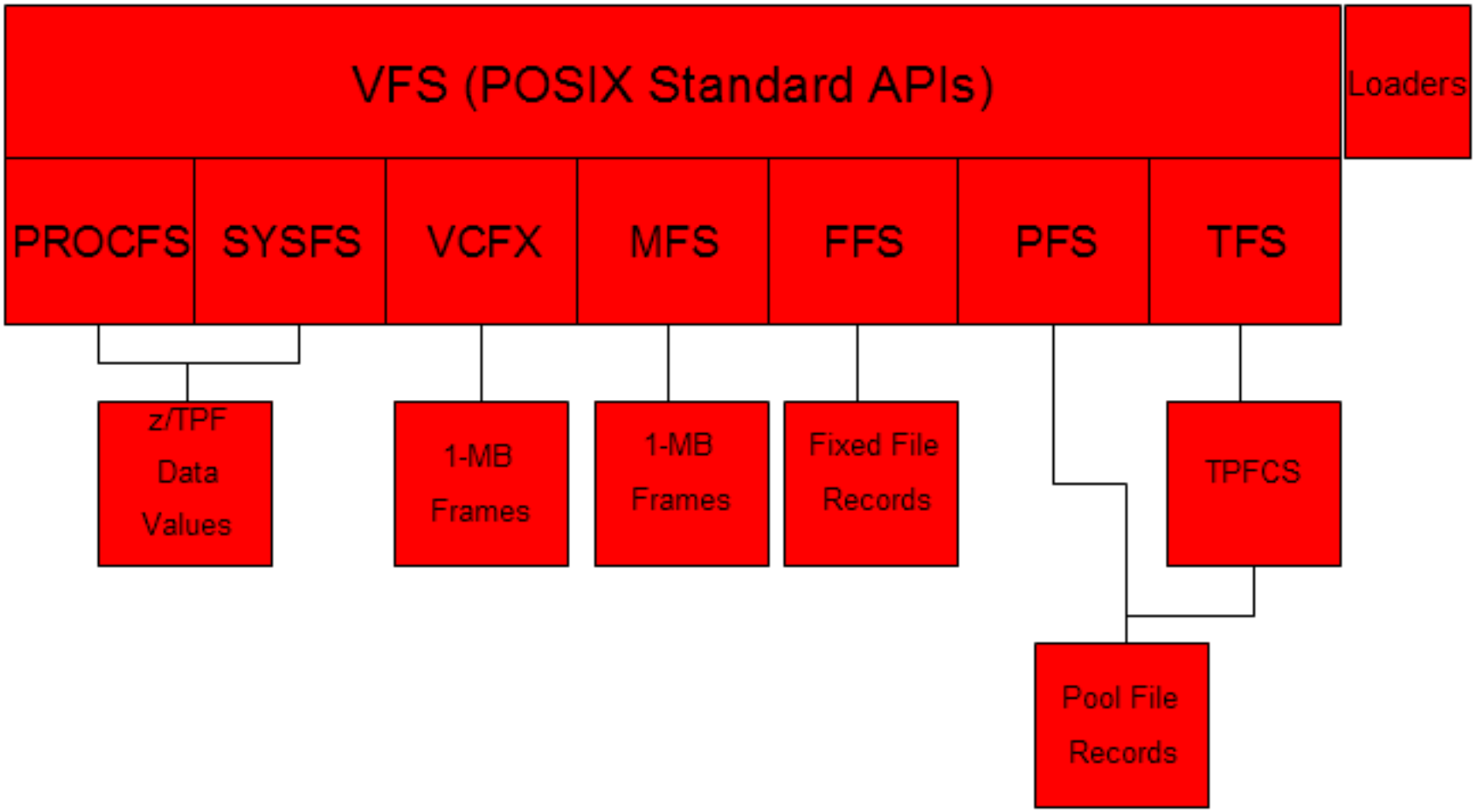


TPF collections support (TPFCS)

- **Benefits**
  - Provides a level of abstraction and easy to understand database APIs and management capabilities
  - Several different ways to organize data (searchable keys, BLOBs, etc.)
  - Shared across loosely coupled processors
- **Drawbacks**
  - TPF-unique API
  - Not easily accessible from remote platform
  - Robust functionality sacrifices access speed



File System

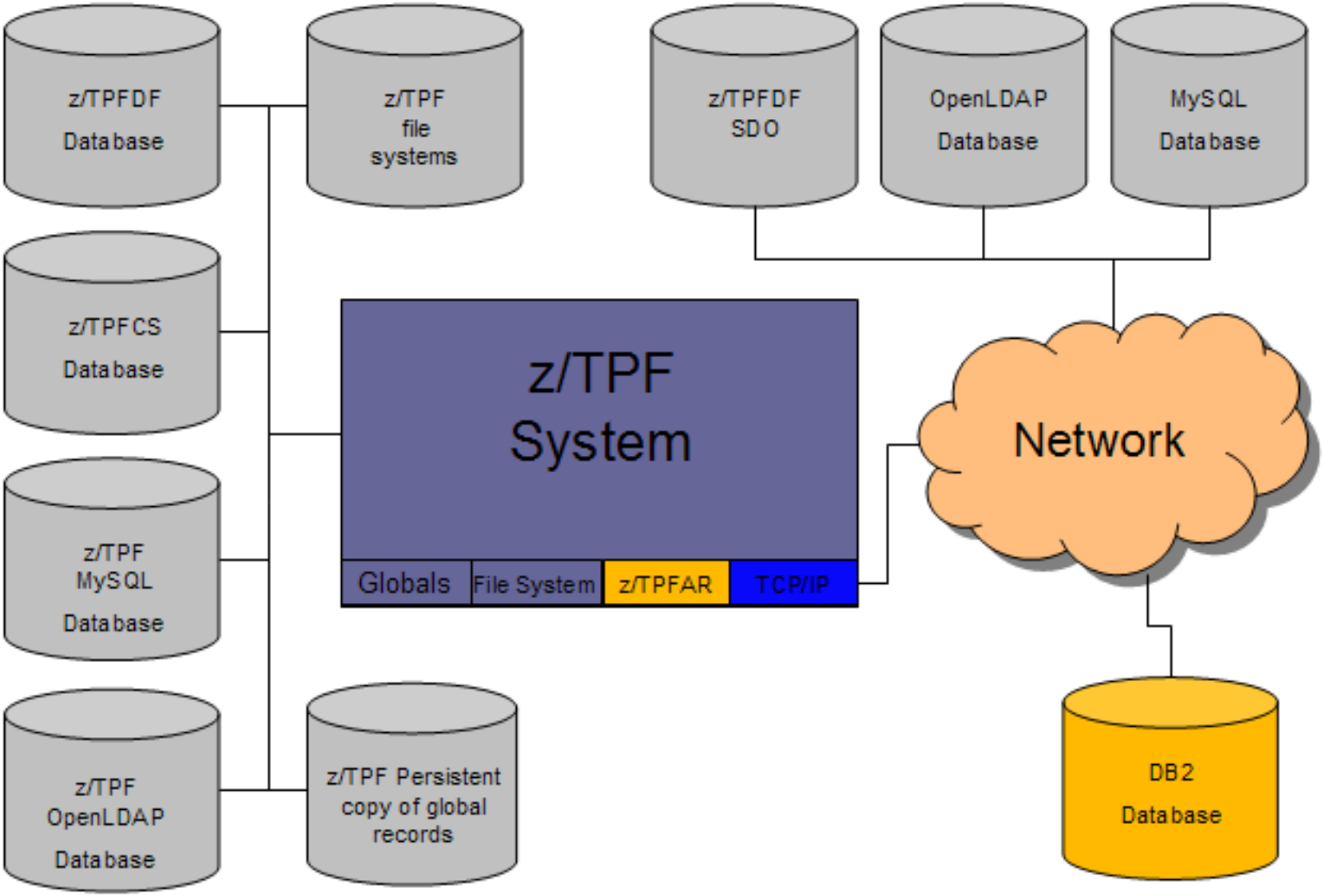


File systems in z/TPF

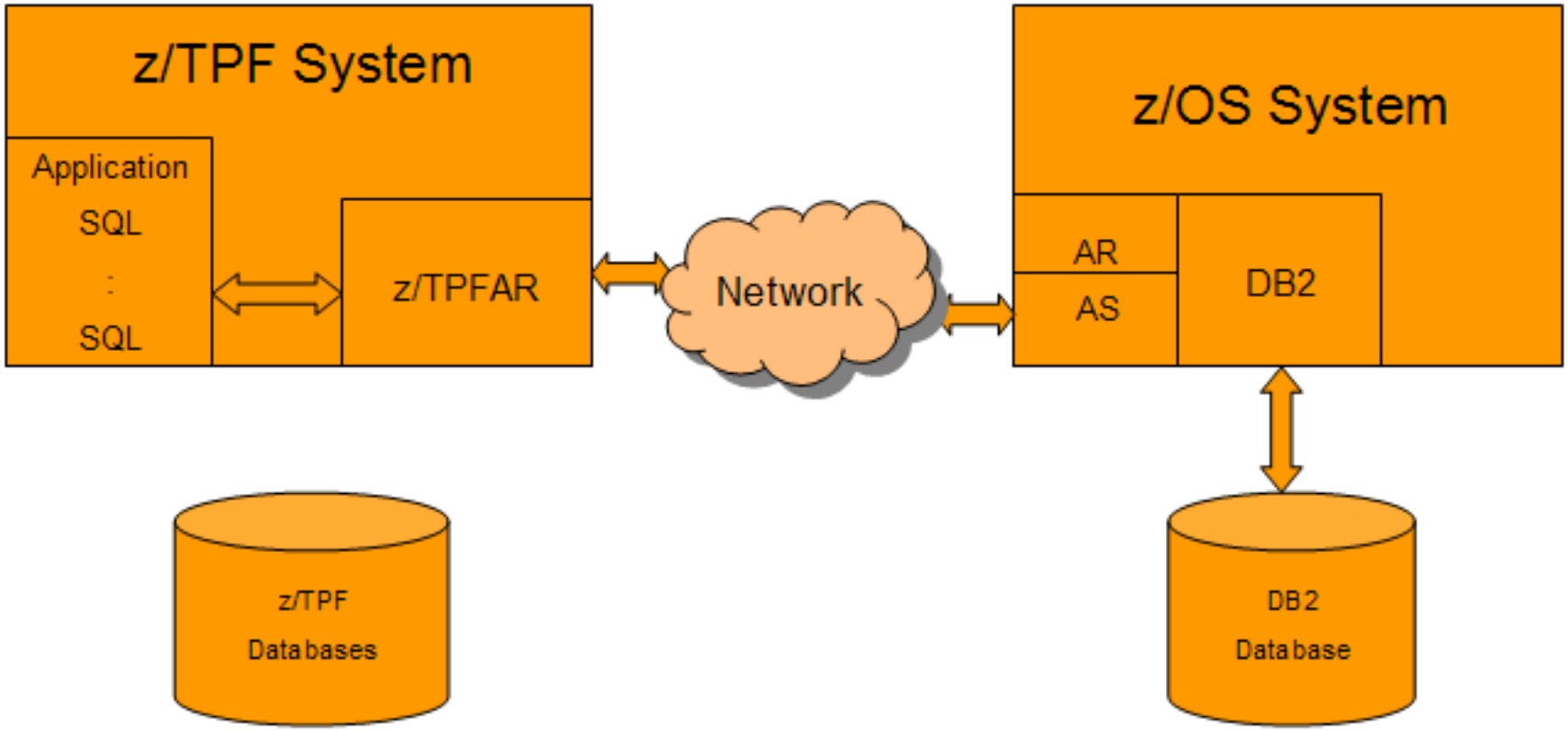
1. TFS - TPF Collection Support File System (renamed original file system in TPF4.1)
   - Processor Shared and Sub-System Unique
   - Root file system
2. MFS - Memory File System
   - Uses system heap as file storage (backed by 1 MB frames)
   - Does not persist over an IPL
   - Processor and Sub-System Unique
3. FFS - Fixed-File File System
   - Uses fixed-file records
   - Persists over an IPL and unmount/mount
   - Processor and Sub-System Unique
4. PFS - Pool File System
   - Uses pool records
   - Persists over an IPL and unmount/mount
   - Processor and Sub-System Unique
5. PROCFS - Process File System
   - **Access to process information**
6. SYSFS - System File System
   - **Access to system information**
7. VCFX - Version Control File System (internal)

File system

- **Benefits**
  - POSIX standard API
  - Virtual file system (VFS) allows for different mountable file system “backends”
    - Processor unique vs. shared
    - Persistent vs. memory-only
  - Files can be loaded like programs using ZOLDR
- **Drawbacks**
  - “Flat files” are limited in functionality:
    - Difficult to structure or organize multiple data entities in a single file
    - Data is not easily searchable
  - Application level needs to do more work to use data compared to solutions built on top of FS



z/TPF application requester (z/TPFAR)



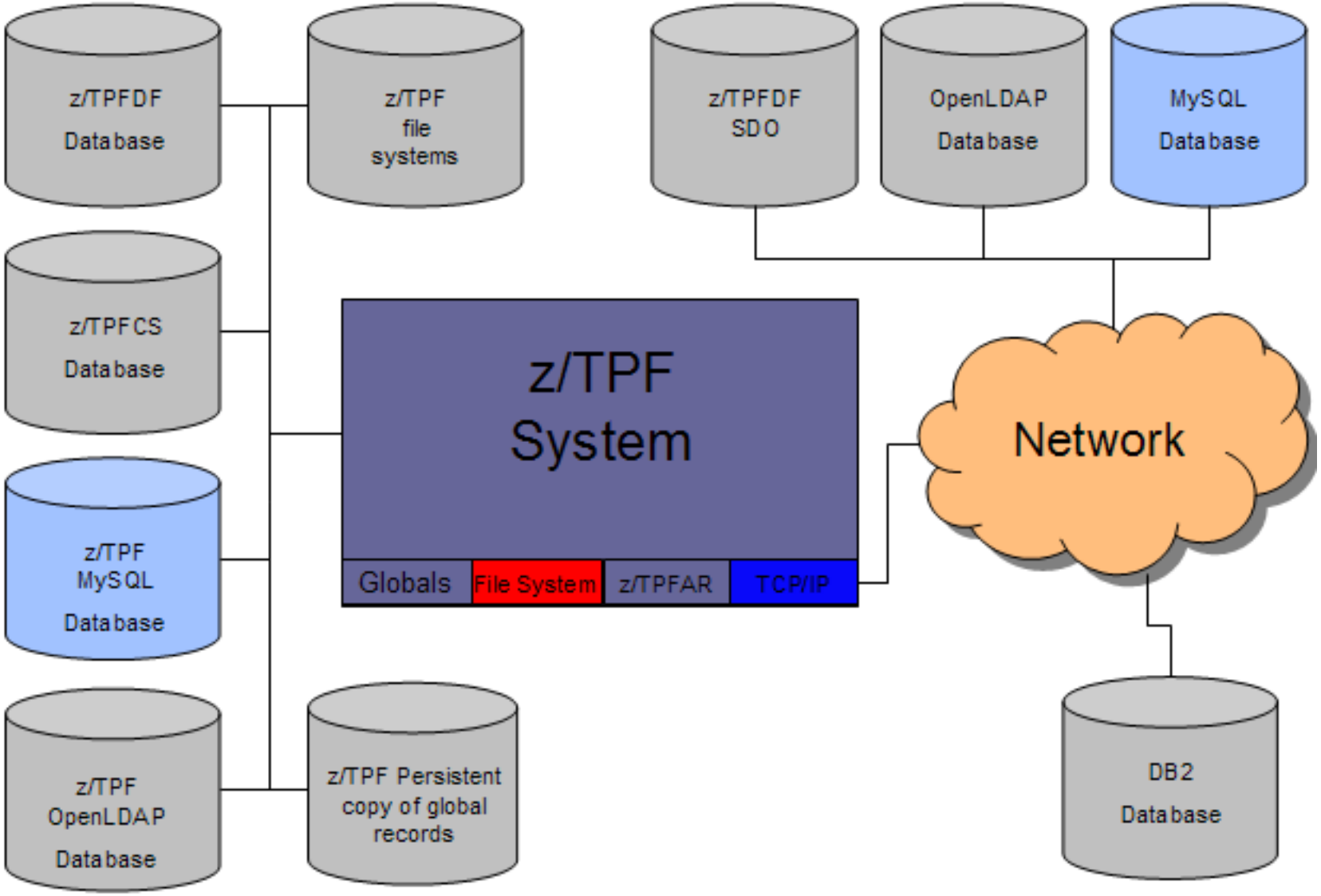
z/TPF application requester (z/TPFAR)

- **Benefits**
  - Provides structured query language (SQL) access to a relational database management system (RDBMS)
  - TPF-unique “hot cons” provides the ability to initiate long-running pipes
- **Drawbacks**
  - Access to data is dependent upon the availability of the remote RDBMS

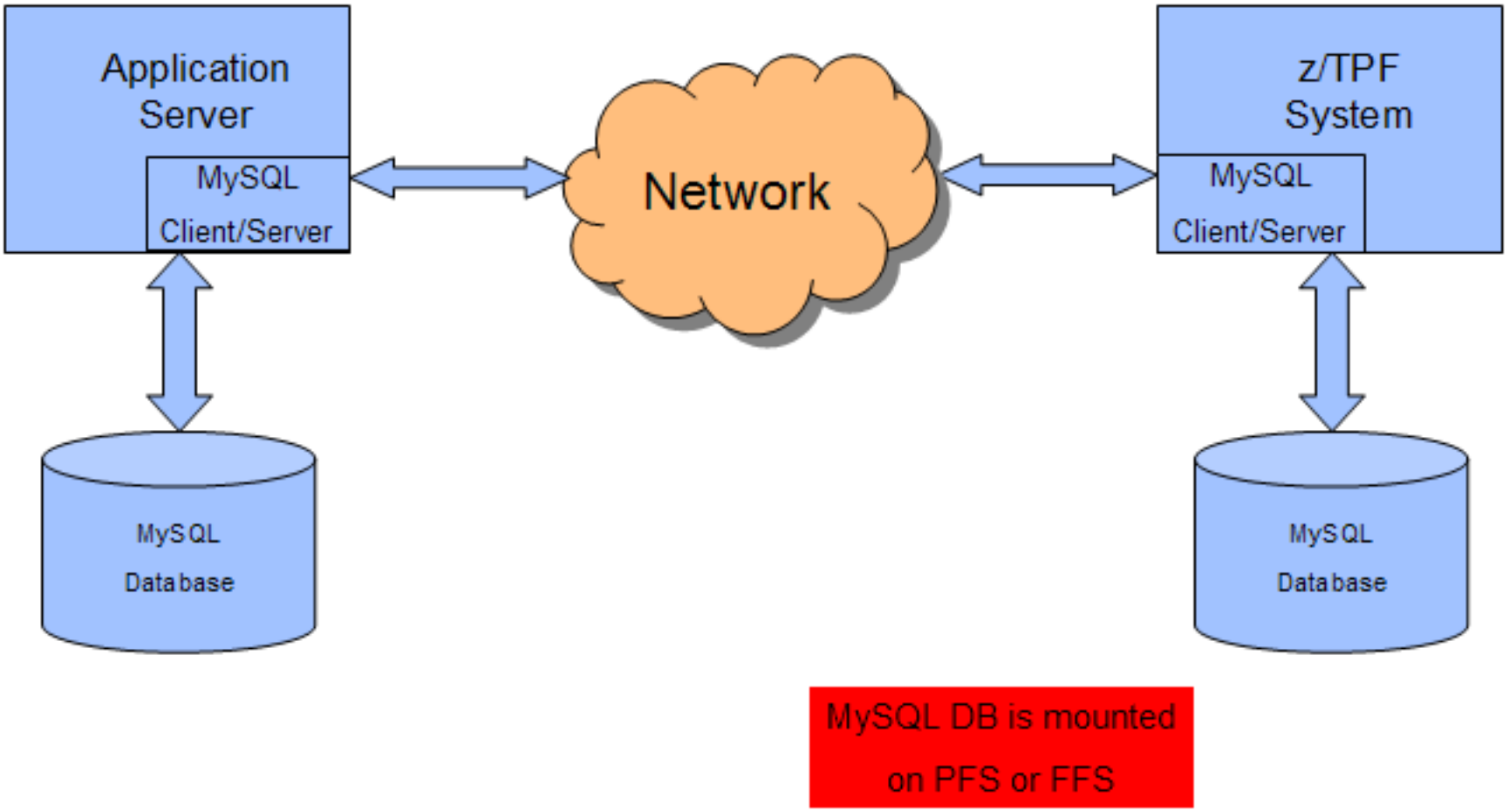
Ported Code Options

The next two options, MySQL and OpenLDAP, were ported to z/TPF. There is an active community who use online web-boards and other mechanisms to communicate. Users will want to utilize this community for expertise in database design and specific tuning parameters. Some examples:

- **OpenLDAP: http://www.openldap.org/lists/mm/listinfo/openldap-software**
- **Oracle BDB: https://forums.oracle.com/forums/forum.jspa?forumID=271**
- **MySQL: http://forums.mysql.com/**

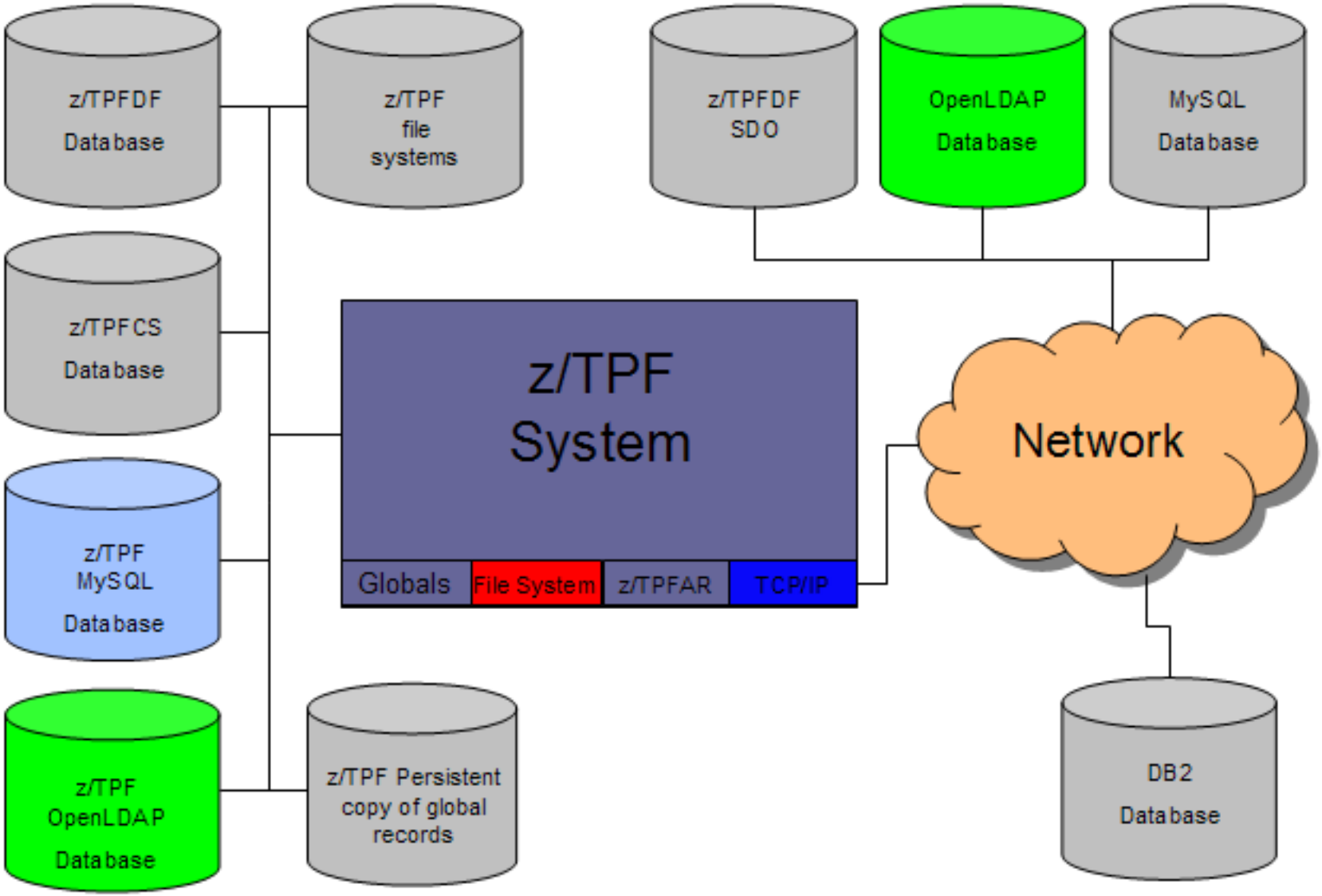


MySQL

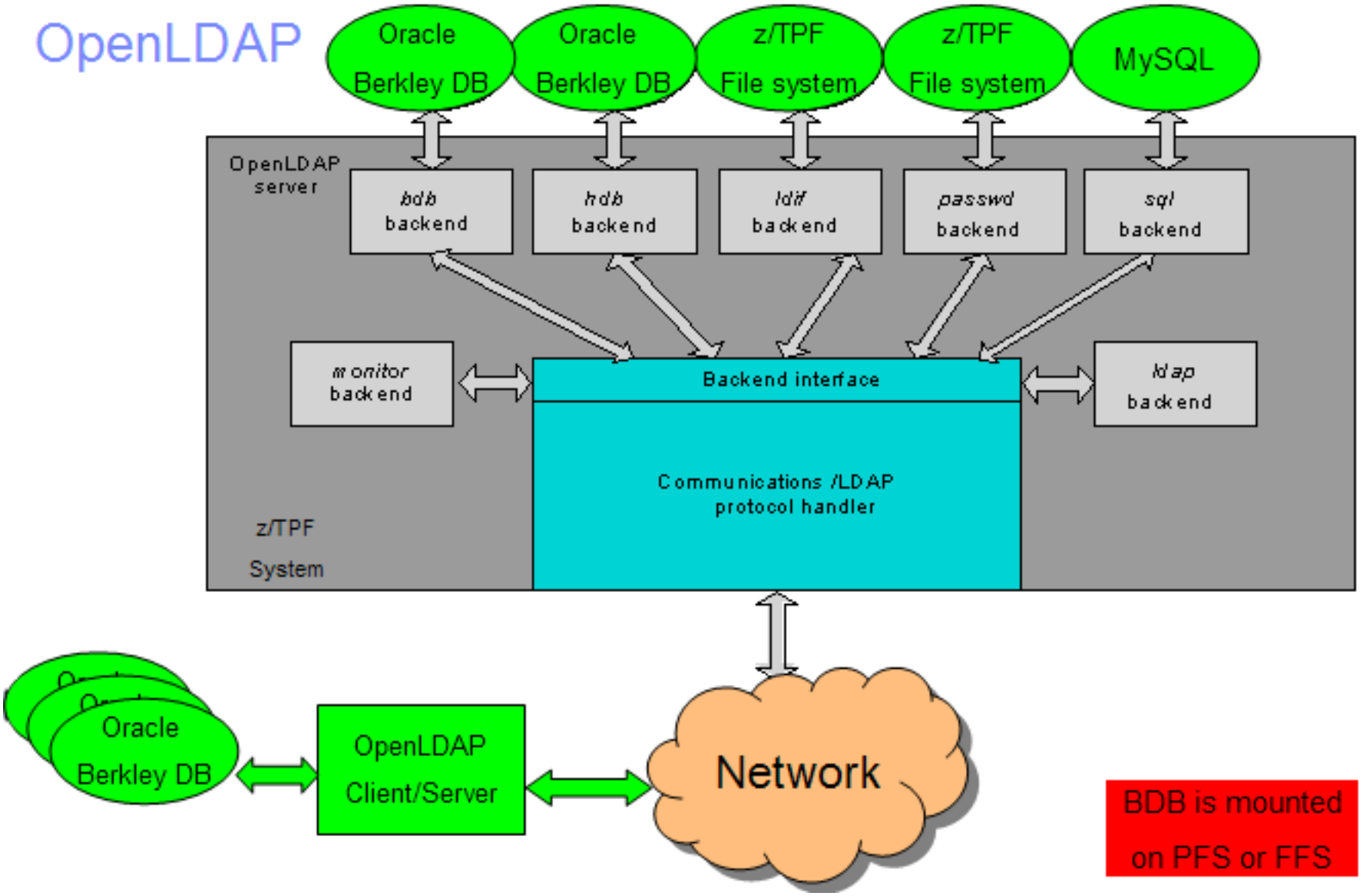


MySQL

- **Benefits**
  - Provides structured query language (SQL) access to a relational database management system (RDBMS)
    - Local RDBMS
    - Remote RDBMS
- **Drawbacks**
  - RDBMS is not optimal from a performance standpoint for high throughput transactional traffic interfacing with large databases
  - Ported to TPF - new versions may be needed in future; database design/tuning requires MySQL expertise.



OpenLDAP



OpenLDAP

- **Benefits**
  - Designed for very efficient read accesses
  - De facto standard API
  - Open source tooling available
  - Client library lets applications access directories on local and remote platforms
- **Drawbacks**
  - “Directory” is not meant to be a complete “database” replacement
  - Not designed for high update rates
  - Ported to TPF - new versions may be needed in future; database design/tuning requires OpenLDAP expertise.

Access to z/TPF data from remote platforms

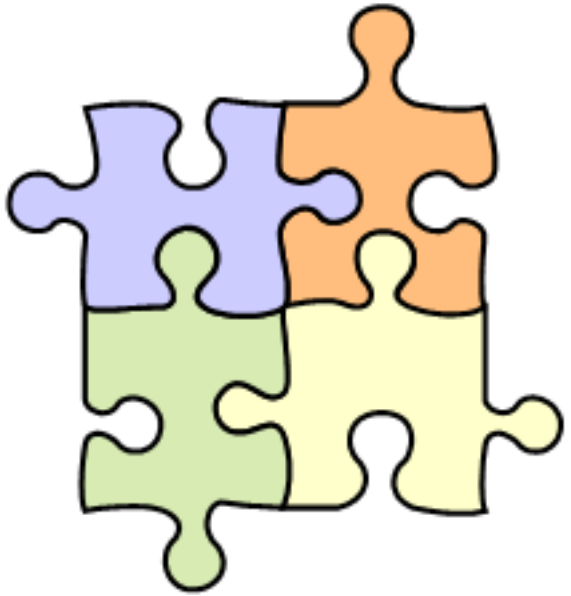
- **Often z/TPF system is the system of record for critical business data**
- **Data owned by z/TPF might be required for use by remote platforms (e.g., data warehousing/analysis)**
- **Several options available to make the data externally viewable:**
  - Direct access via SDO
  - Data replication/transformation performed off z/TPF
  - Data replication/transformation performed on z/TPF

Data management options (comparison summary)

Database Option	On platform API/Off Platform API	Read frequency capabilities	Write frequency capabilities	Administrative processes/costs	Strategic Direction
z/TPF format-1 globals	Non-standard / None	Extremely High	Keypoint: High Synchronized: Low	Manual	No current enhancement plans
z/TPF format-2 globals	Non-standard / None	Extremely High	Keypoint: High Synchronized: Low	Improved over z/TPF format-1 globals	Target of any future globals requirements
z/TPFDF	Non-standard / z/TPFDF SDO	High	High	TPF-unique administrative functions	Primary z/TPF DB for transactional workload
z/TPF Collection Support (z/TPFCS)	Non-standard / None	Medium	Low	Minimal / TPF-unique	No current enhancement plans
File system	Standard / FTP, HTTP	Varied depending on which FS	Varied depending on which FS	Minimal / standard	Usability and remote access enhancements
z/TPF Application Requester (z/TPFAR)	Standard / NA	Low	Low	Standard (DB administration off platform)	No current enhancement plans
MySQL	Standard / Standard	Medium-High	Low-Medium	Standard	Ongoing support
OpenLDAP	Standard / Standard	Medium-High	Low	Standard	Ongoing support

Does IBM recommend any options?

I believe everyone knows the answer
IT DEPENDS!
This presentation was intended to address some of the issues involved in making the decision.



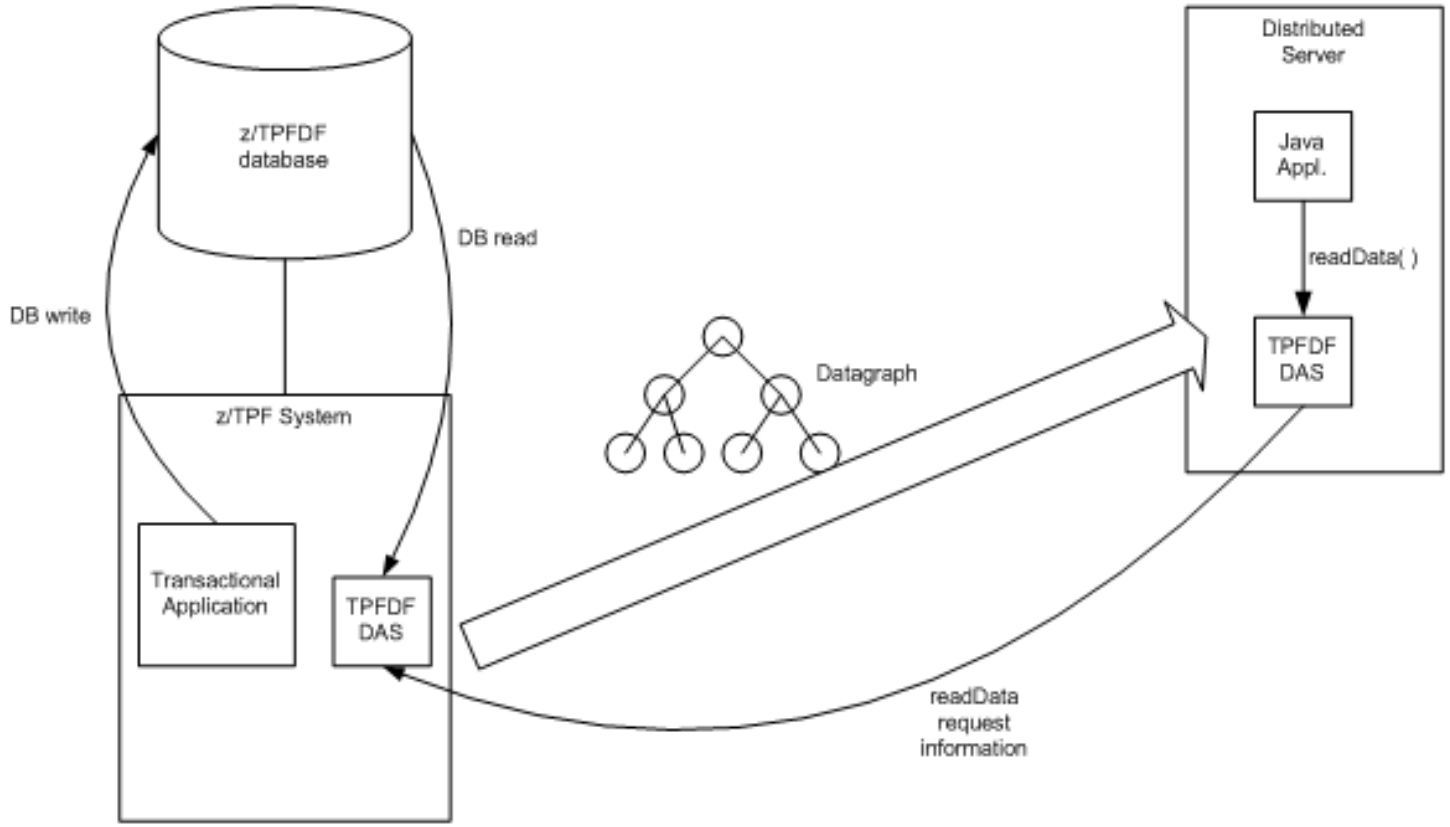
Trademarks

- IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- OpenLDAP is a registered trademark of the OpenLDAP Foundation.
- Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
- MySQL® and the MySQL logo are registered trademarks of MySQL AB.
- Other company, product, or service names may be trademarks or service marks of others.

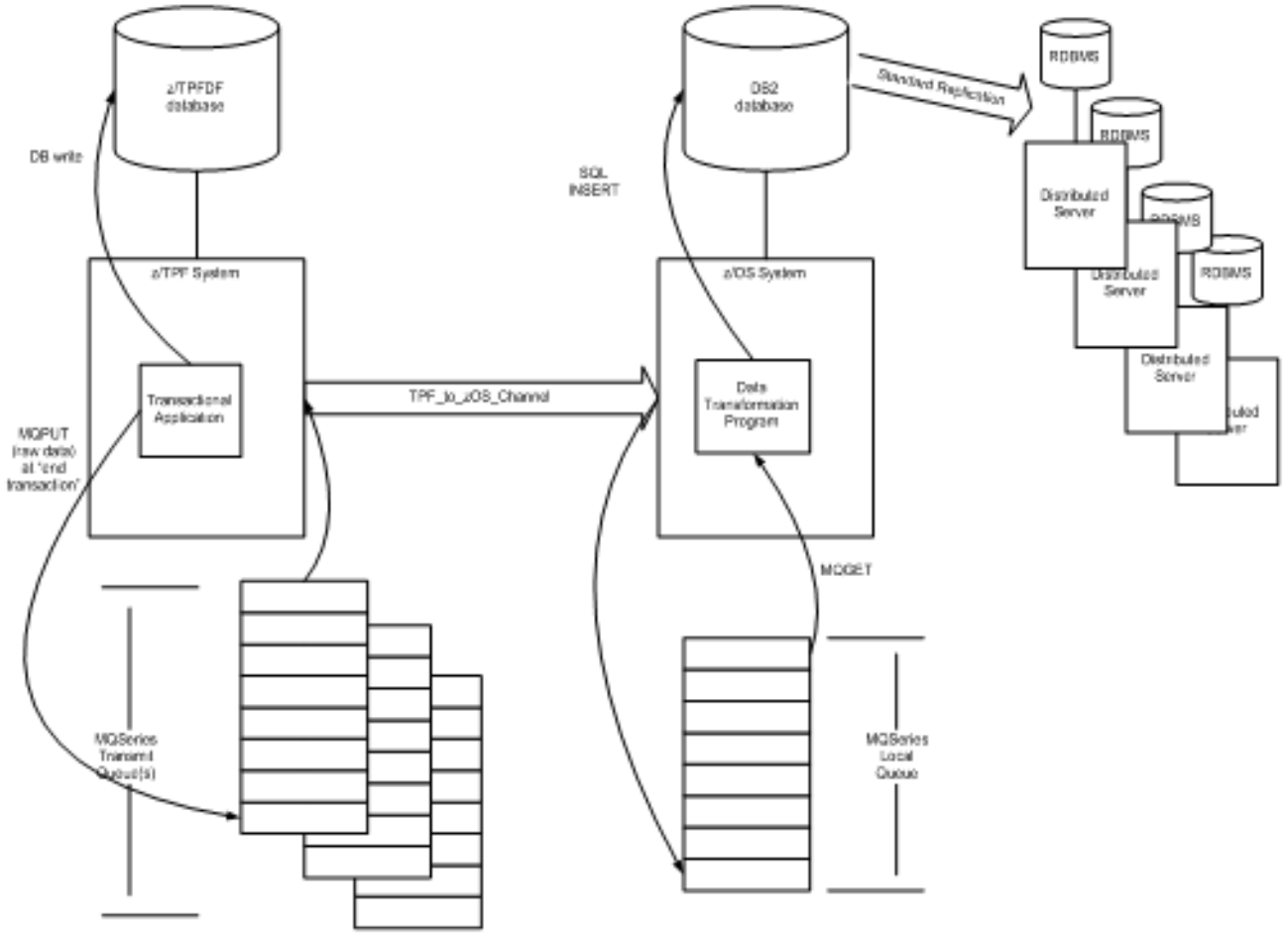
- Notes
- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
- This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

EXTRA SLIDES

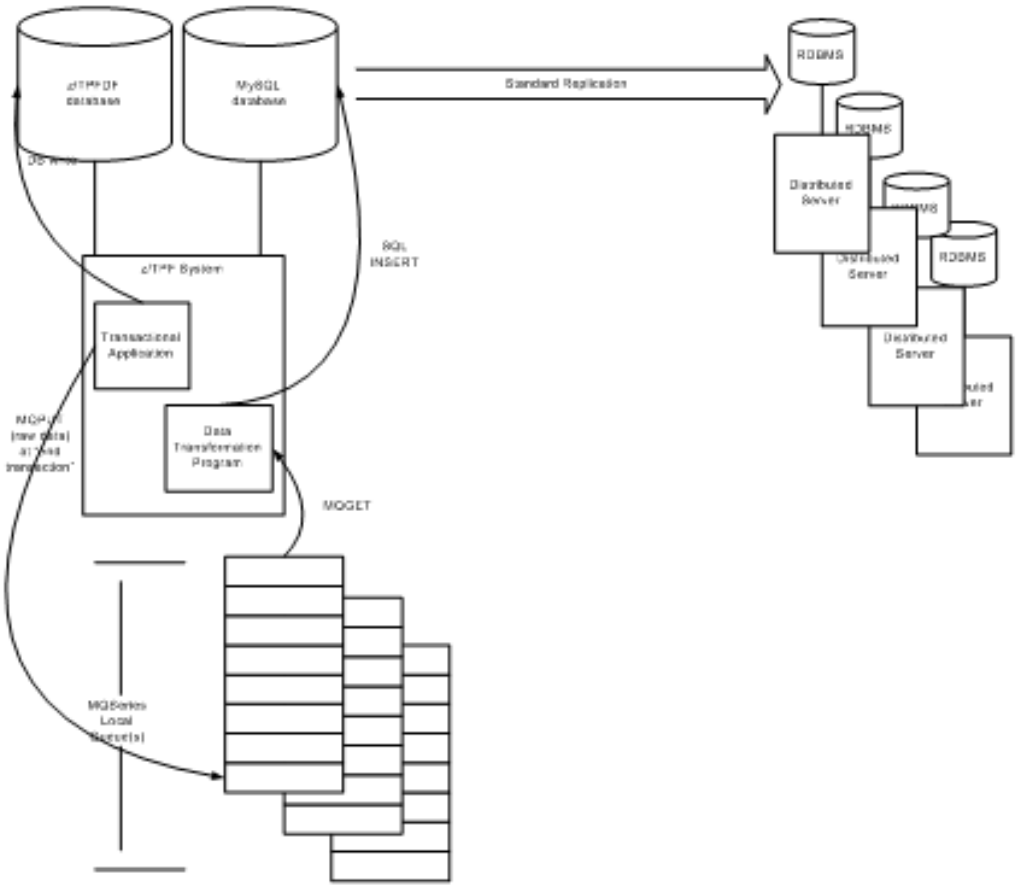
Direct data access via z/TPFDF SDO



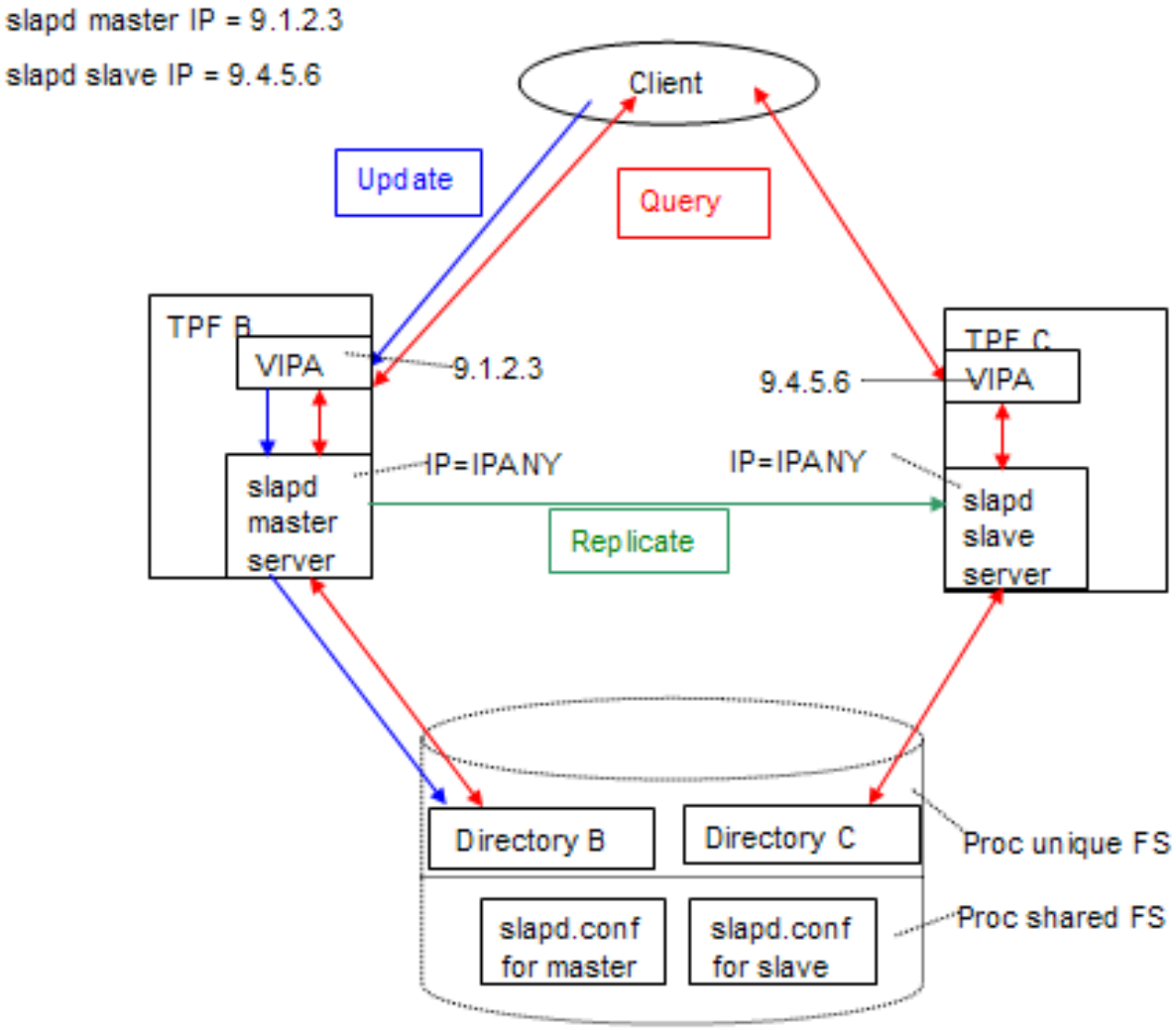
Data replication off platform via WebSphere MQ



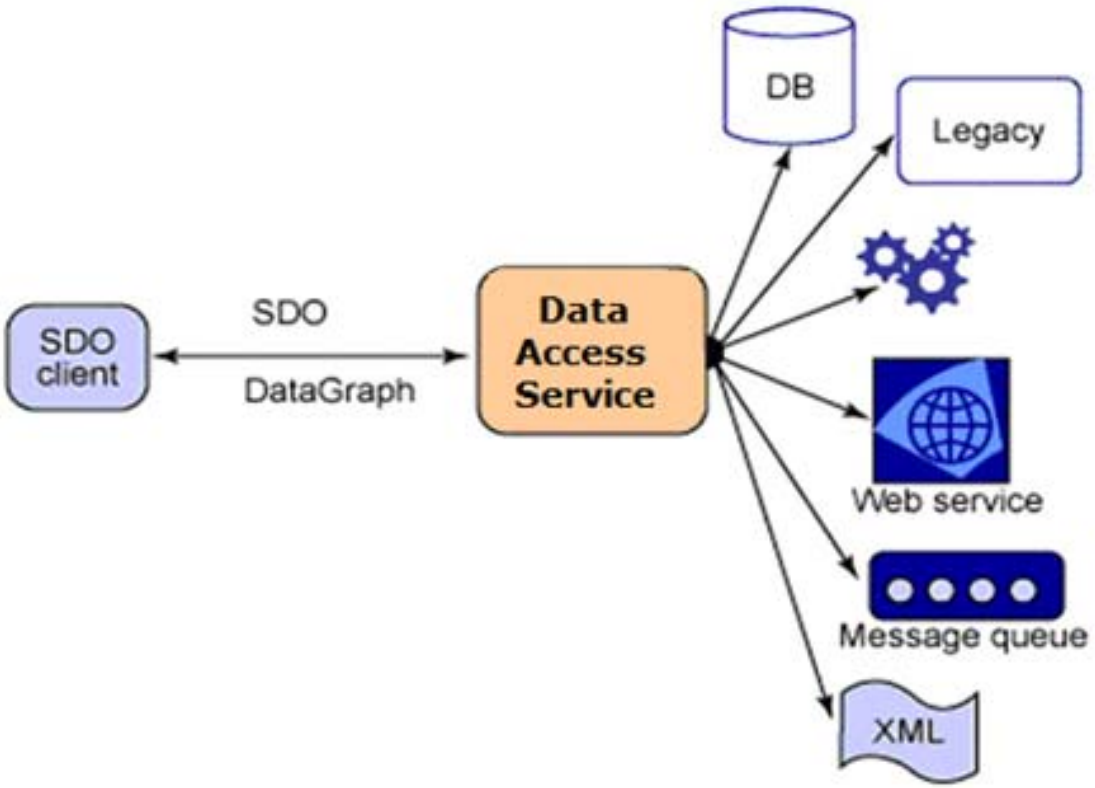
Data replication with MySQL on z/TPF



OpenLDAP availability using Loosely Coupled support



SDO: Architectural diagram



SDO: Terminology and concepts

- **DataObject – entity representing fragment of data**
  - Property (single-valued, many-valued, “simple”, “complex”)
  - Type (String, Integer, Date, Boolean, DataObject)
  - z/TPFDF
    - DataObject – subfile, LREC
    - Property – field in LREC, reference (link) from LREC to another subfile
- **DataGraph – graph representing data (non-persistent)**
  - Tree of DataObjects

SDO: Data access service (DAS)

- **Data Access Service (DAS)**
  - Specific form of SCA (Service Component Architecture) service
  - Load DataGraph from a data source or service, for example
    - XML data sources – XML file DAS
    - Relational databases – JDBC DAS
  - Propagate changes back into the data source
  - Disconnected model