

Developing Samba VFS Modules

Richard Sharpe
DELL & Samba Team
rsharpe@samba.org

Agenda

- Overview of VFS Modules
- The Life Cycle of a Module
- VFS Support Functions
- Writing Your Module
- Building and Debugging
- More Information

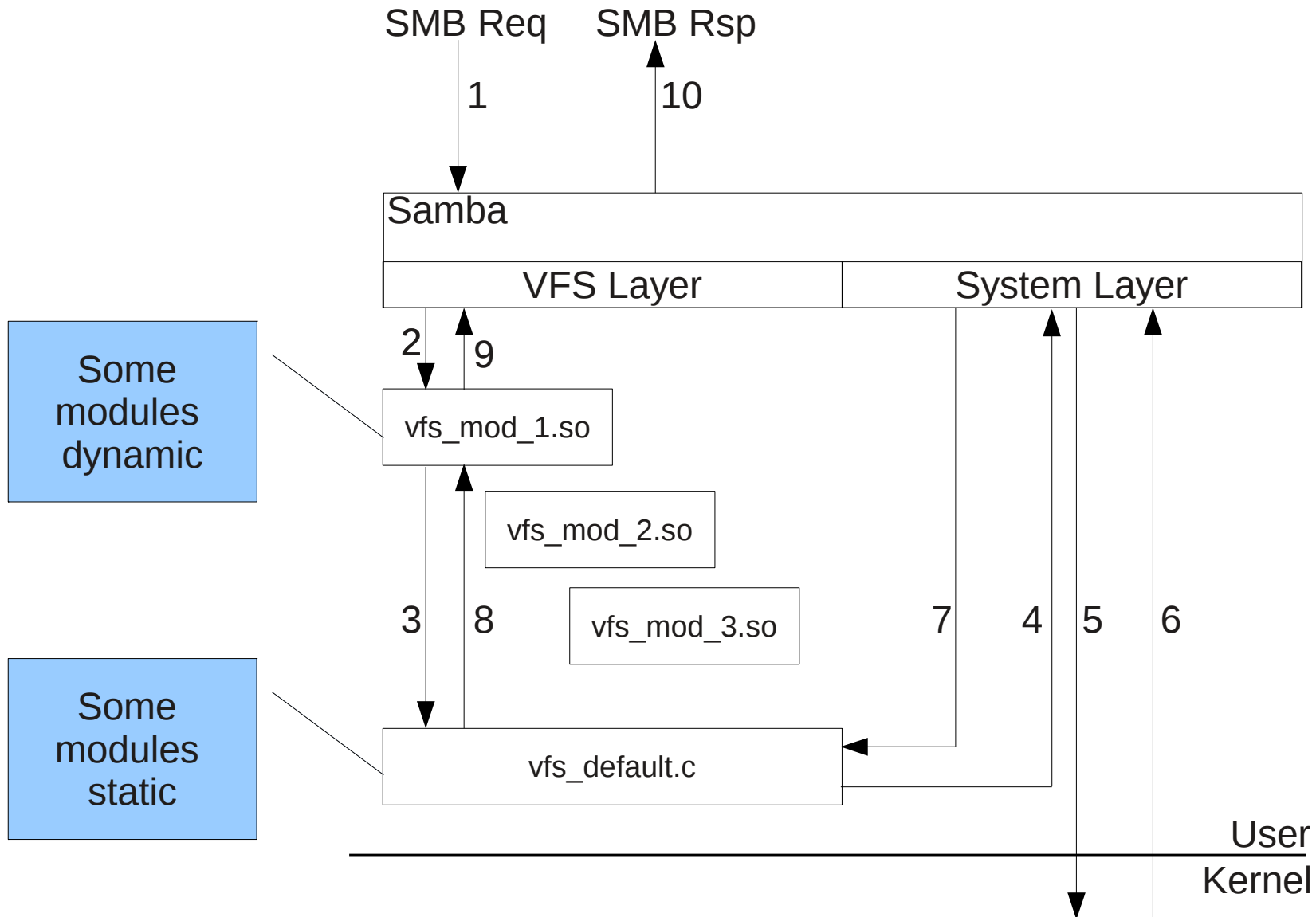
Overview of VFS Modules

- What do they do?
- Where do they fit?
- Different types of modules
- The interface: Classes of functions & functions
- What does the default module do?
- Some existing modules
- Samba 3.5.x vs 3.6.x vs master
- Why?

What do they do?

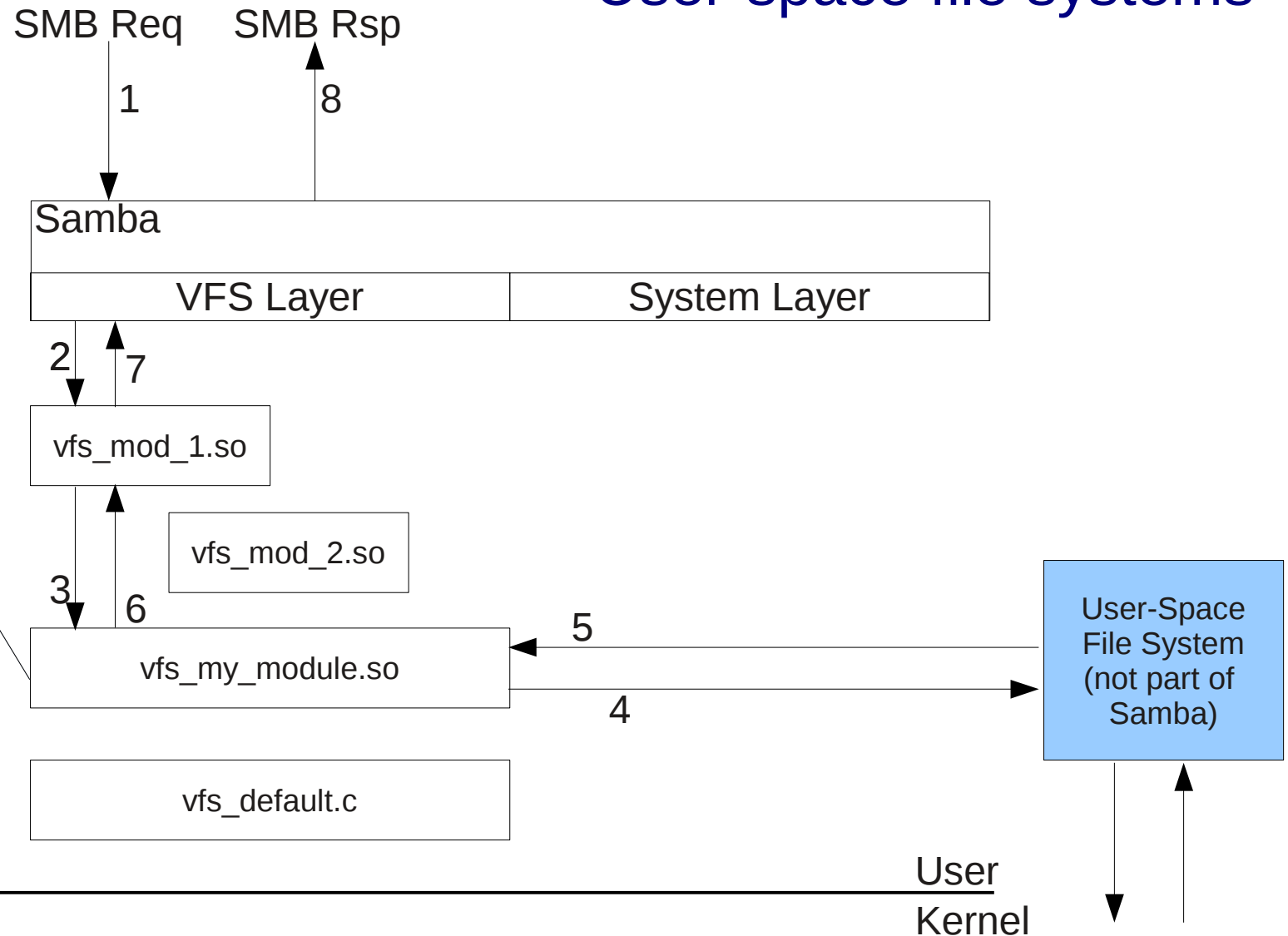
- Extend Samba's functionality
 - NT ACL support in XATTRs or TDBs
 - Interface to user-space file systems
 - Ceph via libceph
 - Provide access to functionality of special file systems
 - Convert NT ACLs to NFSv4 ACLs on GPFS
- Many existing VFS modules
 - You can add your own

Where do they fit



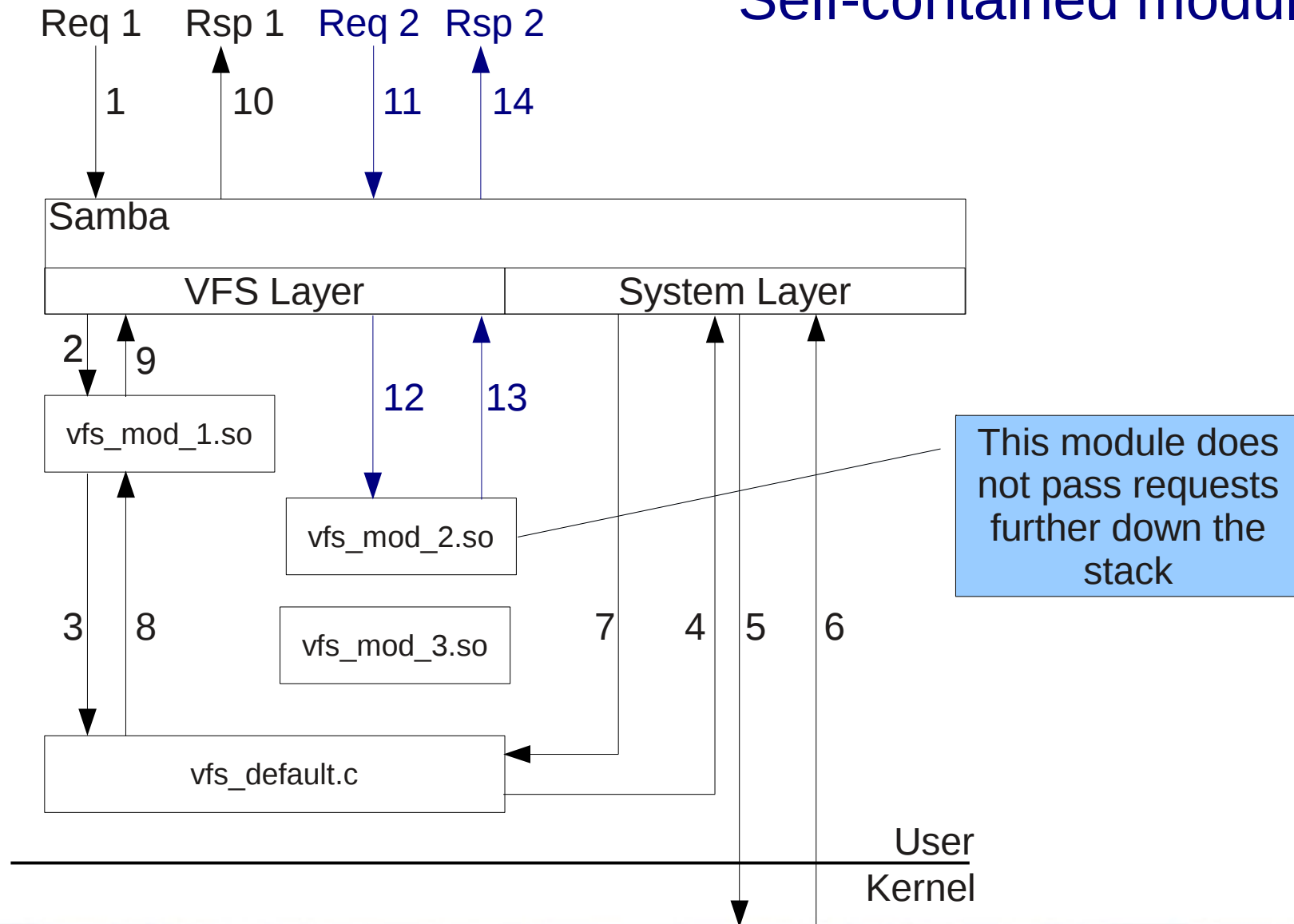
Different types of modules

User-space file systems



Different types of modules, cont

Self-contained module(s)



The interface: Classes of functions and functions

- 8 classes
 - Disk/share/fs operations
 - Directory operations
 - File operations
 - NT ACL operations
 - POSIX ACL operations
 - EA operations
 - AIO operations
 - Offline operations

The interface: Classes of functions and functions, cont

- connect_fn, disconnect_fn
- open_fn, create_file_fn, read_fn, write_fn, ...
- fget_nt_acl_fn, get_nt_acl_fn, ...
- More than 100 functions

The interface: Notes

- Paths passed in usually relative to the root of the share
 - So, you can simply concatenate root and path
- Might not be the case going forward
 - Track chdir changes with `chdir_fn`

What does the default module do

- Provides default functionality
 - Calls into Samba proper
 - Back-stops all other modules
 - Call it if you are extending existing functions
 - Don't call it if you are replacing existing functions
 - **Don't call it if your file descriptors are unknown by the kernel!**

Some existing modules

- `samba-source/source3/modules`
 - `acl_xattr.so, vfs_tdb.so`
 - `gpfs.so`
 - `ceph.so`
 - `readahead.so, readonly.so`
 - `etc`

Samba 3.5.x vs 3.6.x vs master

- Master
 - All functions called <fn-name>_fn
 - At least two new functions
 - FSCTL & AUDIT_FILE
- 3.6 & 3.5
 - Fewer functions (some protos changed in 3.5 to 3.6 transition)
 - Inconsistent function naming, many without _fn

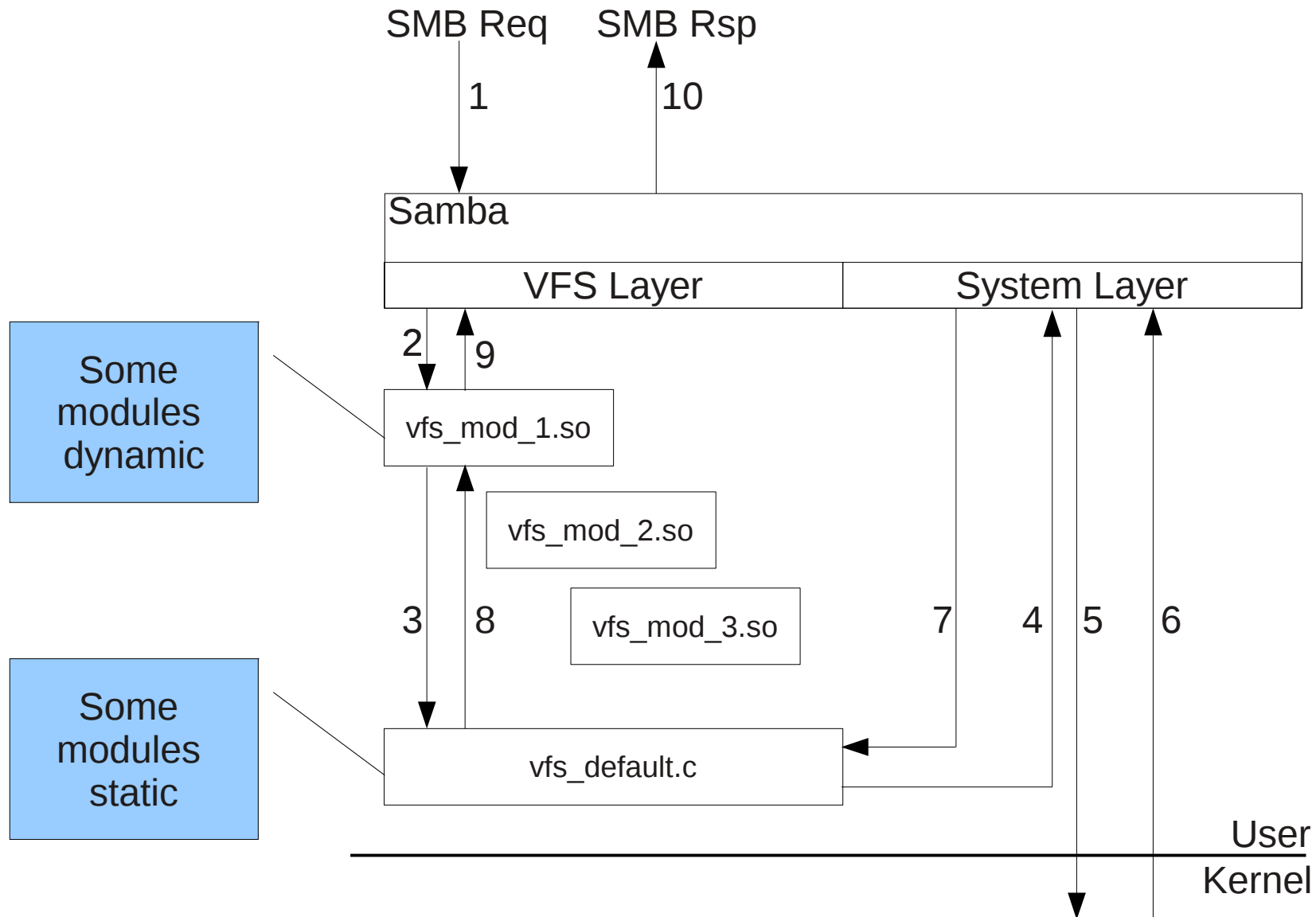
Why?

- If you want Samba changes
 - You can isolate your changes
 - Easier to move to new versions of Samba
- Add functionality

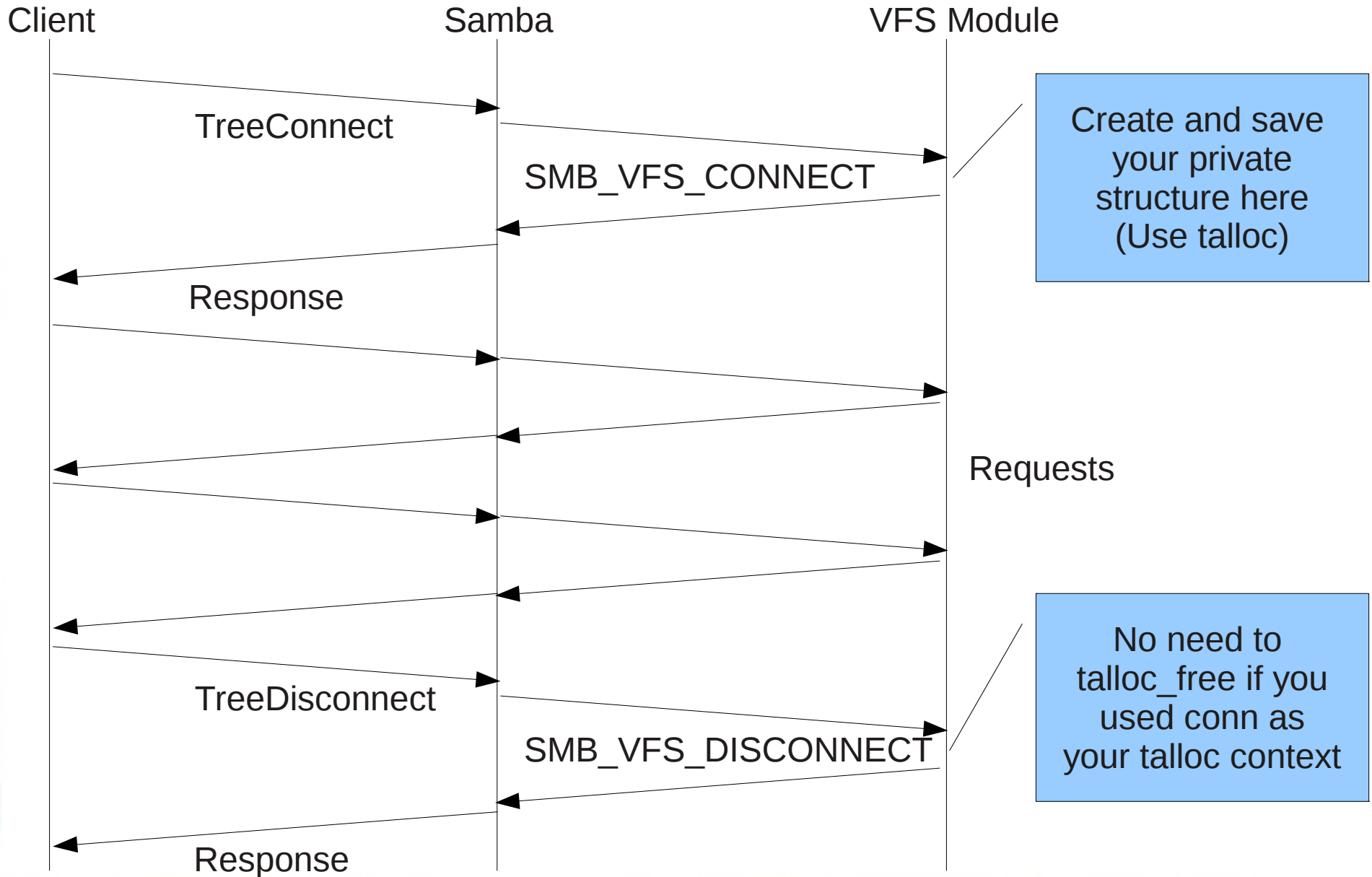
VFS Module Details

- Life cycle overview
- Look at real code
 - `vfs_demo`
 - `vfs_amazon_s3`
 - `vfs_acl_xxx`
 - Supporting code in Samba

Putting it together



Life cycle overview



Handling your private data

```
ctx = talloc_zero(handle, struct demo_struct);
if (!ctx) {
    DEBUG(10, ("some-error"));
    return -1;
}
SMB_VFS_HANDLE_SET_DATA(handle, ctx, NULL,
                        struct demo_struct,
                        goto error);

return 0;
error: some-error-statements;
```

Handling your private data, cont

- Retrieving your private data

```
SMB_VFS_HANDLE_GET_DATA(handle,  
                          ctx,  
                          struct demo_struct,  
                          goto ctx_error);
```

Extending FSPs

- Adding extensions to files ...

```
VFS_ADD_FSP_EXTENSION(handle,  
                        fsp,  
                        struct my_struct,  
                        NULL);
```

Writing your module

- View code from here on

Building and debugging

- View code here
 - Building out of tree ... [examples/VFS](#)
 - Debugging hints ...

More Information

- [Http://samba.org/~sharpe/](http://samba.org/~sharpe/)
 - [The-Samba-VFS.pdf](#)
- [samba-source/examples/VFS](#)
- [samba-source/source3/modules/*.c](#)
- [git@github.com:RichardSharpe/demo-samba-vfs.git](https://github.com/RichardSharpe/demo-samba-vfs)
- [git@github.com:RichardSharpe/Amazon-S3-VFS.git](https://github.com/RichardSharpe/Amazon-S3-VFS)