Apache Web Server

SOFTWARE ARCHITECTURE

ERFAN SHARAFZADEH

Background

▶ The Apache HTTP Server ("httpd") was launched in 1995

Web server developers: Market share of the top million busiest sites
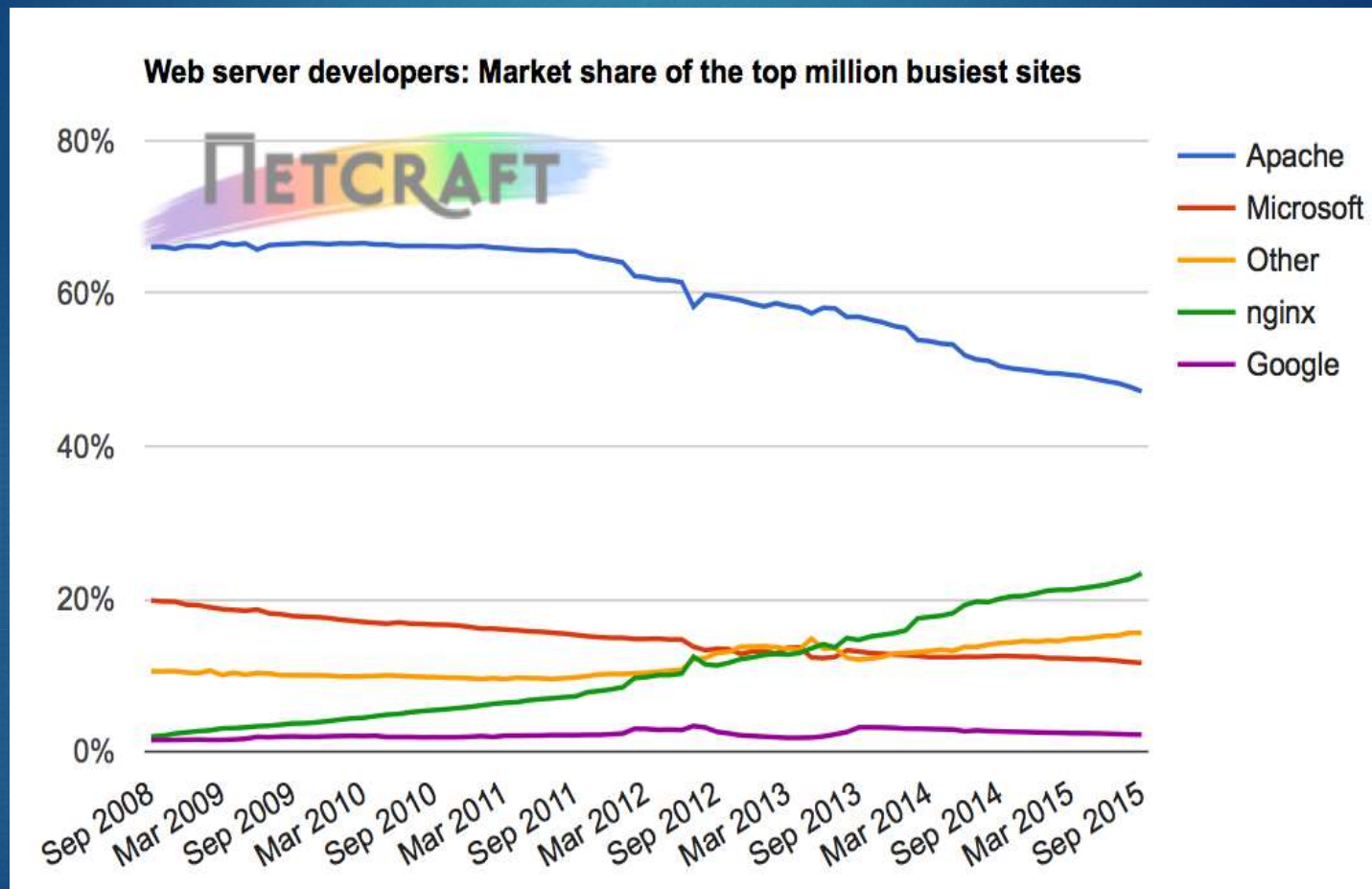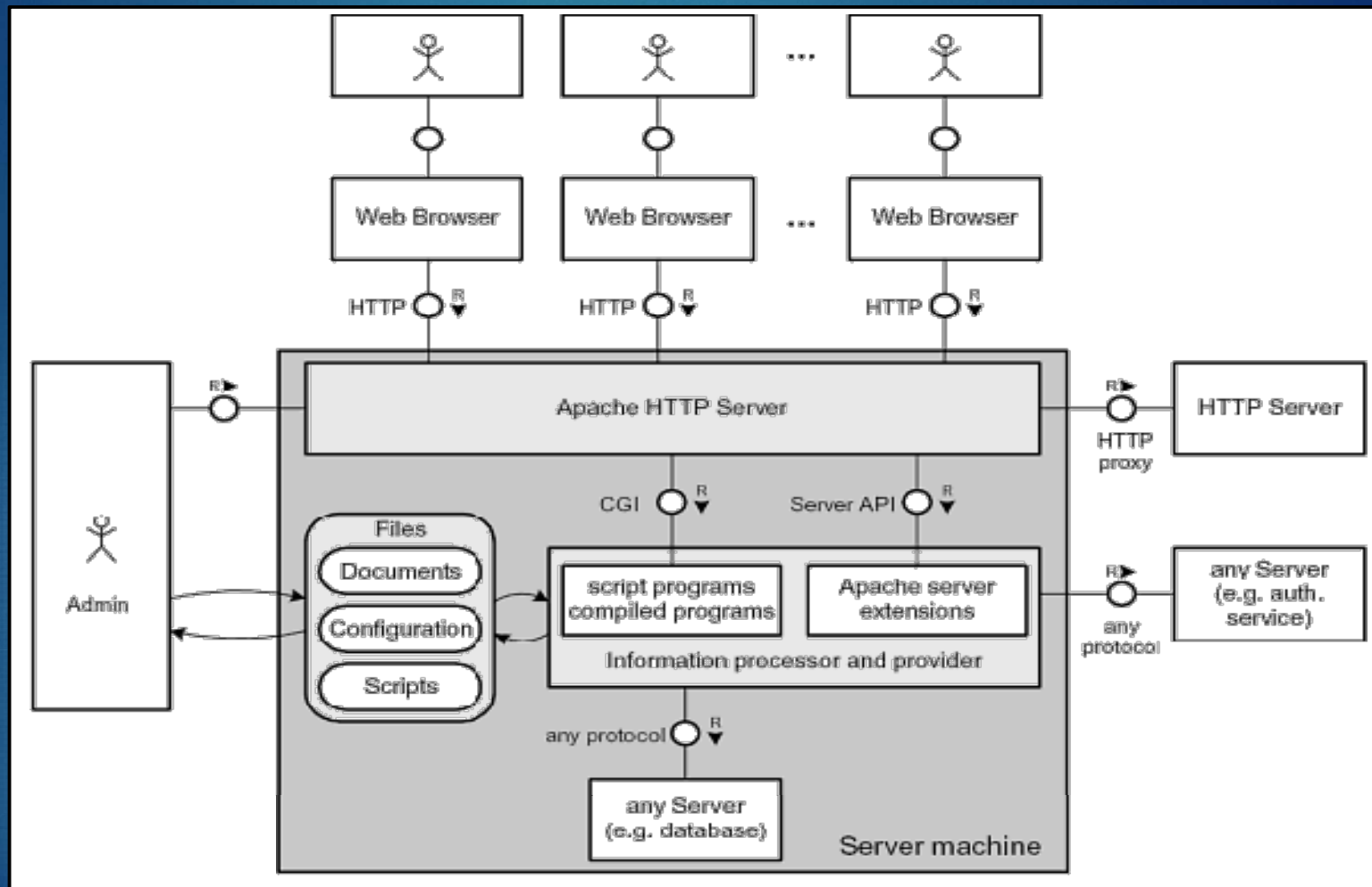
From Monolithic...   ...to Dynamic

Three-tier, J2EE-style architectures → Microservices

Complex protocols (HTML, SOAP) → Lightweight (REST, Messaging)

Persistent deployments → Containers, VMs

Fixed, static Infrastructure → SDN, NFV, Cloud

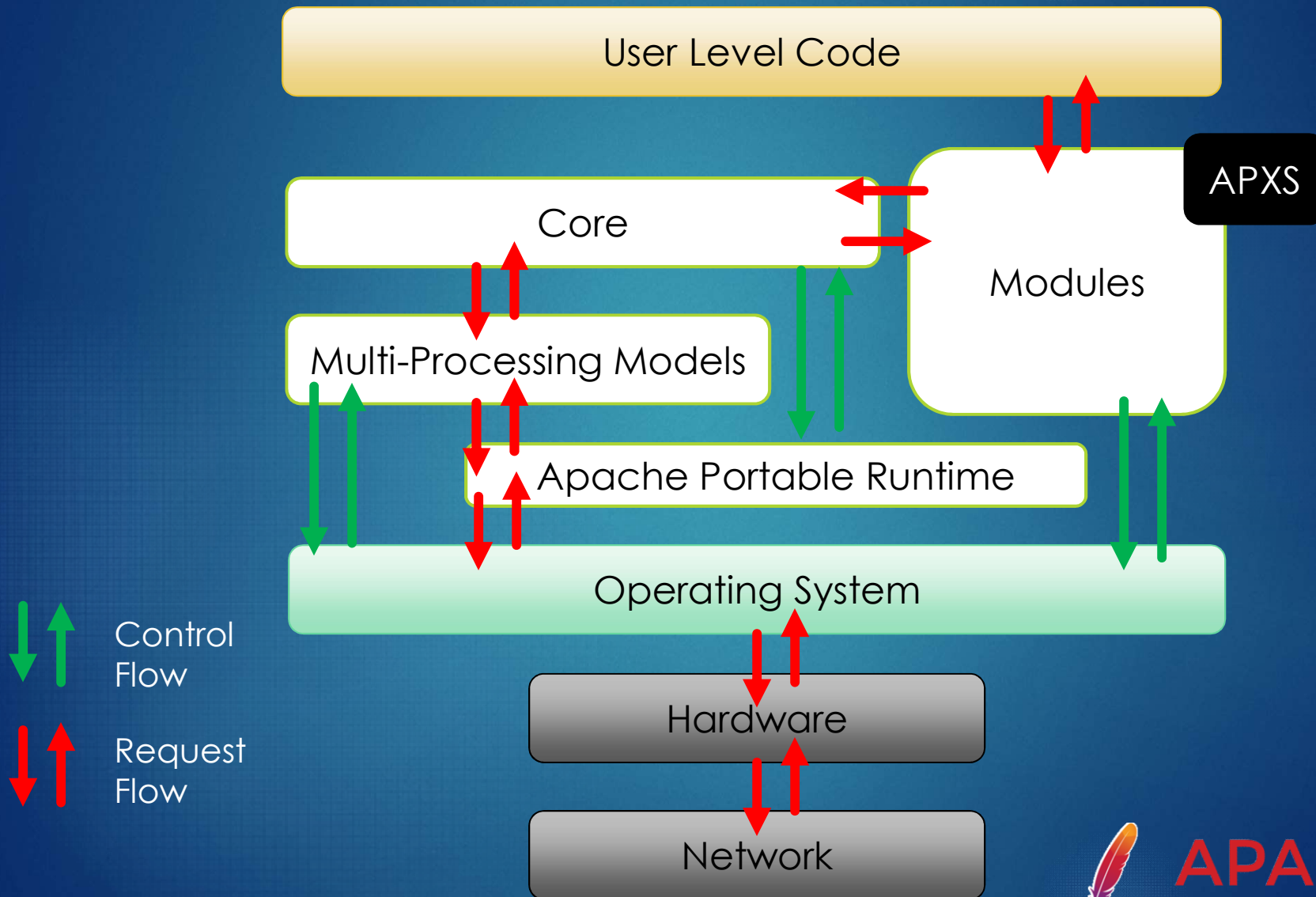Big-bang releases → Continuous delivery

Silo'ed teams (Dev, Test, Ops) → DevOps Culture

APACHE
HTTP SERVER PROJECT

Abstractions

User Level Code

APXS

Core

Modules

Multi-Processing Models

Apache Portable Runtime

Operating System

Control Flow

Request Flow

Hardware

Network

Erfan Sharafzadeh – Software Architecture Course Presentation

APACHE
HTTP SERVER PROJECT

Project Structure

```
                    ┌─────────┐
                    │  HTTPd  │
                    └────┬────┘
   ┌──────┬───────┬──────┼───────┬──────┬────────┐
┌──┴──┐ ┌─┴────┐ ┌┴────┐ ┌┴─────┐ ┌┴──────┐ ┌┴───┐ ┌─┴──────┐
│ OS  │ │Include│ │Build│ │Server│ │Support│ │Test│ │Modules │
└─────┘ └──────┘ └─────┘ └──────┘ └───────┘ └────┘ └────────┘
```

APACHE
HTTP SERVER PROJECT

Project Structure

```
                        HTTPd

  OS    Include   Build   Server   Support   Test   Modules
```
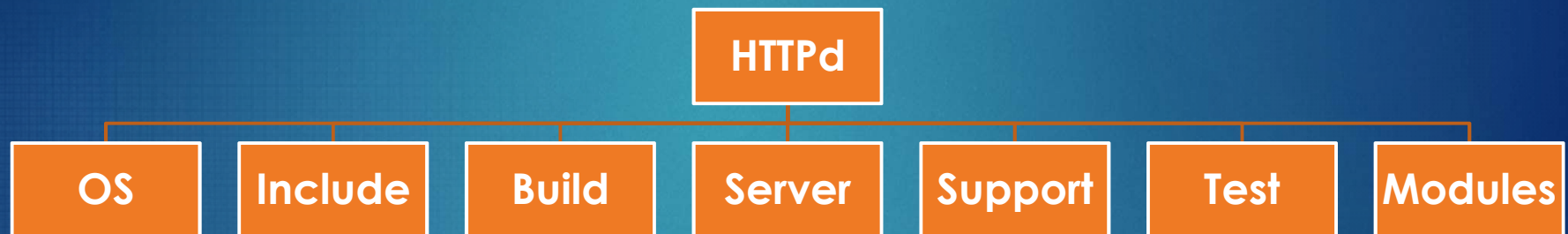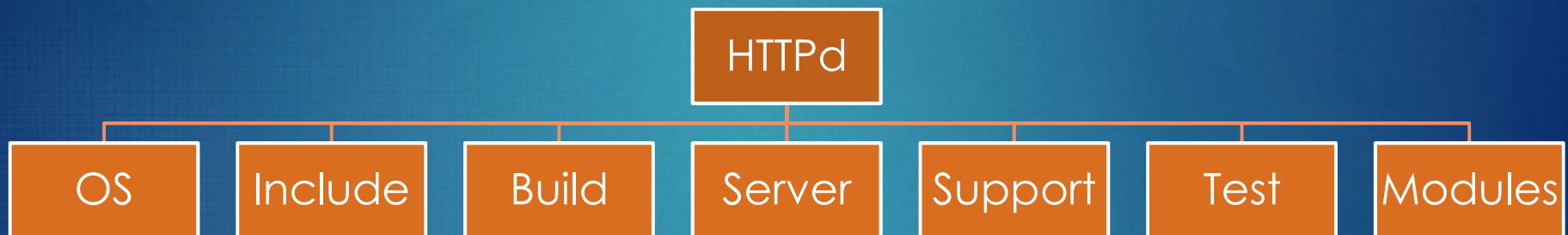
Platform Specific build libraries

Erfan Sharafzadeh – Software Architecture Course Presentation

APACHE
HTTP SERVER PROJECT

Project Structure

HTTPd

| OS | Include | Build | Server | Support | Test | Modules |
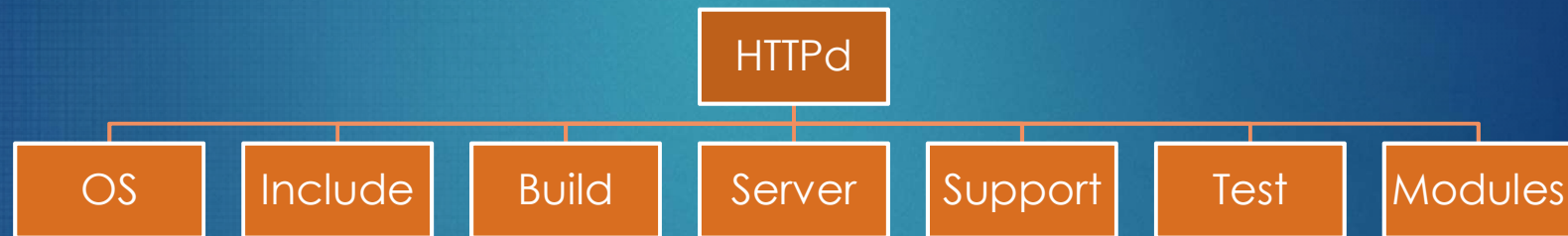
Apache Server Header Files

Erfan Sharafzadeh – Software Architecture Course Presentation

APACHE
HTTP SERVER PROJECT

Project Structure

HTTPd

Modules | OS | Include | Build | Server | Support | Test

Core

Multiprocessing Models

Core Server Functionality

OS interfaces for processing requests

Erfan Sharafzadeh – Software Architecture Course Presentation

APACHE
HTTP SERVER PROJECT

Project Structure

```
                        HTTPd
   ┌──────┬──────┬──────┼──────┬──────┬──────┐
 Modules   OS   Include  Build  Server Support Test
   │
  AAA
   │
  Cache
   │
 Database
   │
  Cluster
   │
  HTTP
   │
  SSL
   │
  ...
```

Modules are mandatory or optional according to the use case

Erfan Sharafzadeh – Software Architecture Course Presentation

APACHE
HTTP SERVER PROJECT

Connection Handling

HTTPd

Modules **OS** **Include** **Build** **Server** **Support** **Test**

Core **Multiprocessing Models**

Prefork

Worker

Event

Netware

Win NT
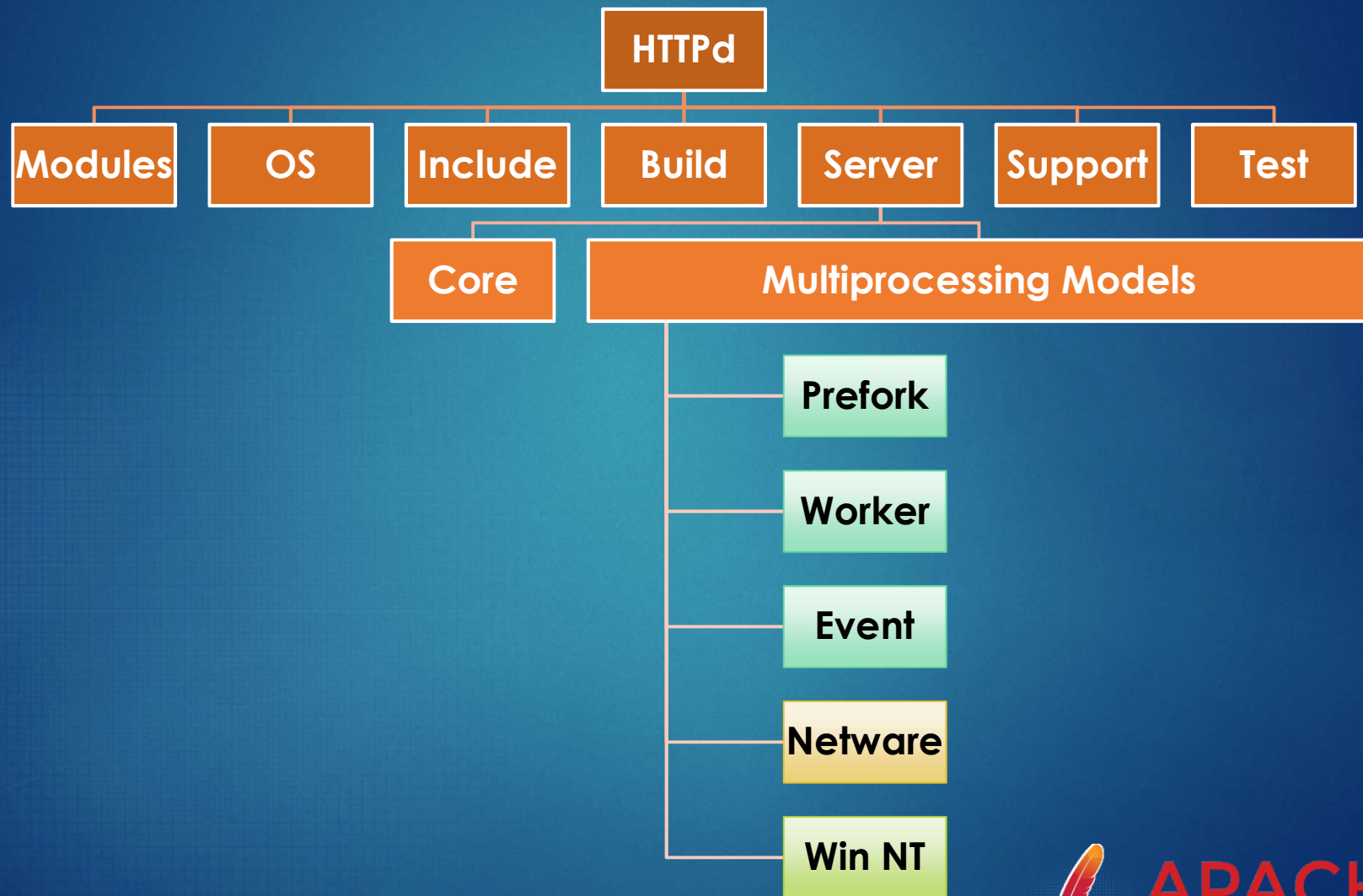
APACHE
HTTP SERVER PROJECT

▶ **mpm_prefork**   ▶ **mpm_worker**   ▶ **mpm_event**
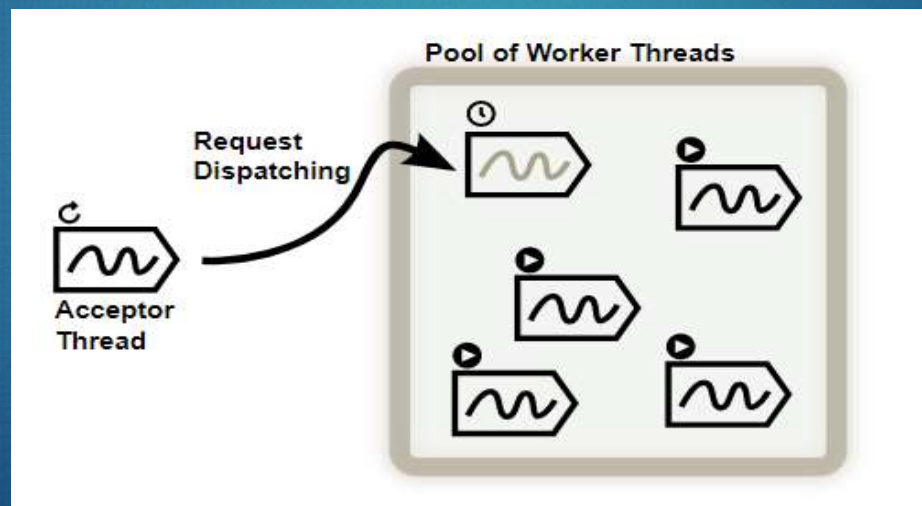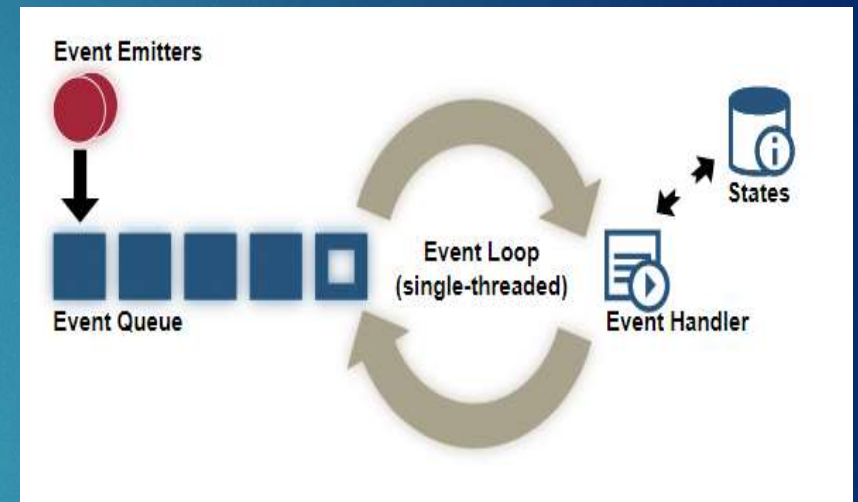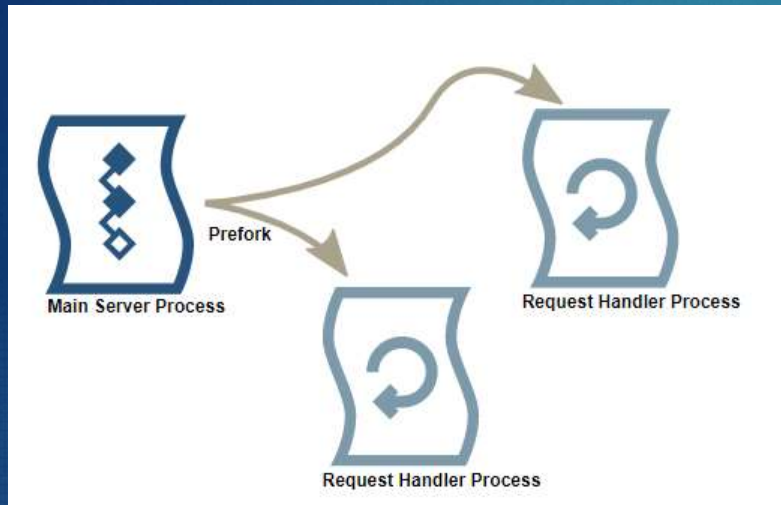
Lets take a look at some animations

https://www.datadoghq.com/blog/monitoring-apache-web-server-performance/
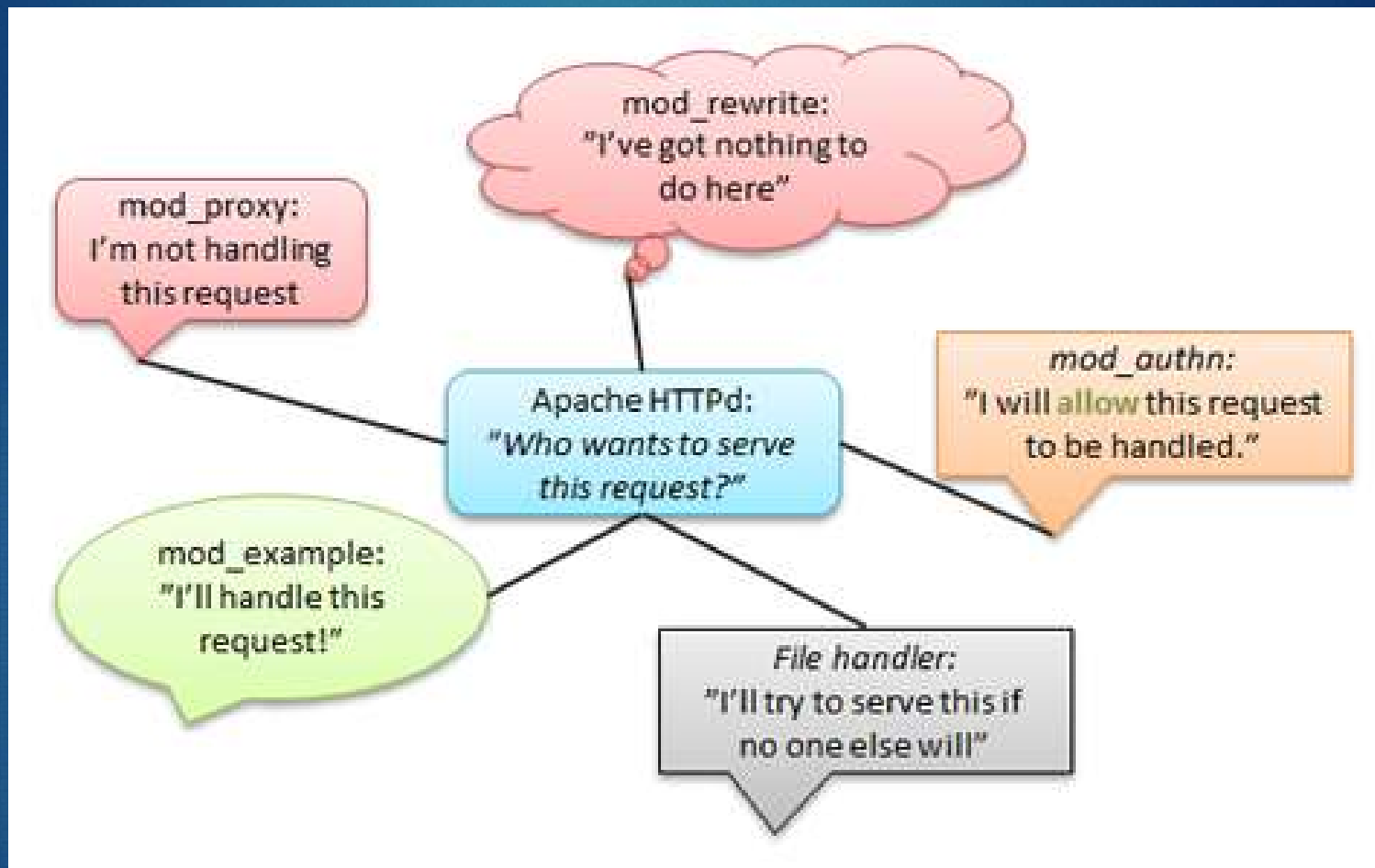
APACHE
HTTP SERVER PROJECT

Modules Interface

Load Module (x) → Register_hooks() → Sort_hooks()

How Modules Work

Erfan Sharafzadeh – Software Architecture Course Presentation

Erfan Sharafzadeh – Software Architecture Course Presentation

APACHE
HTTP SERVER PROJECT
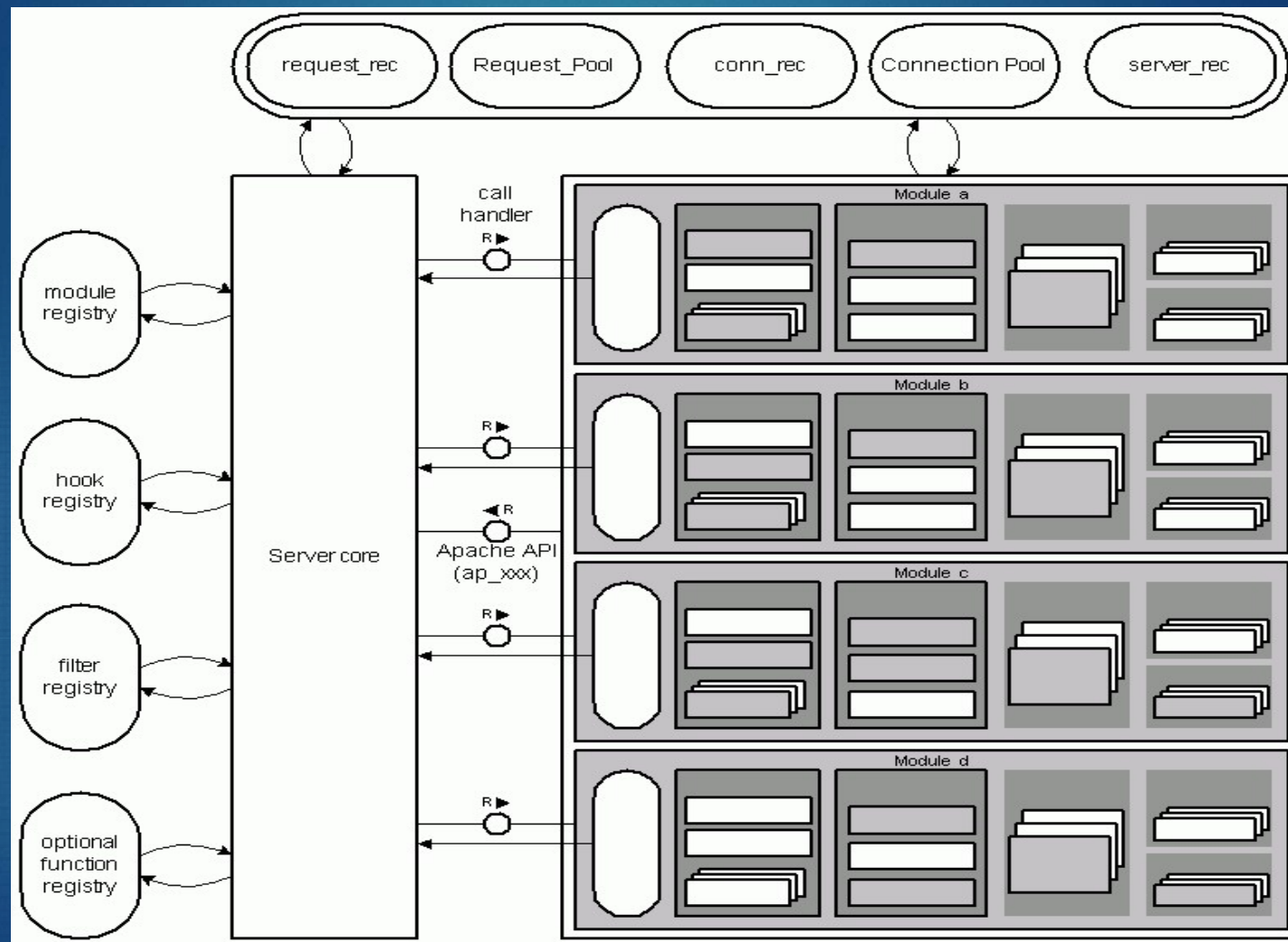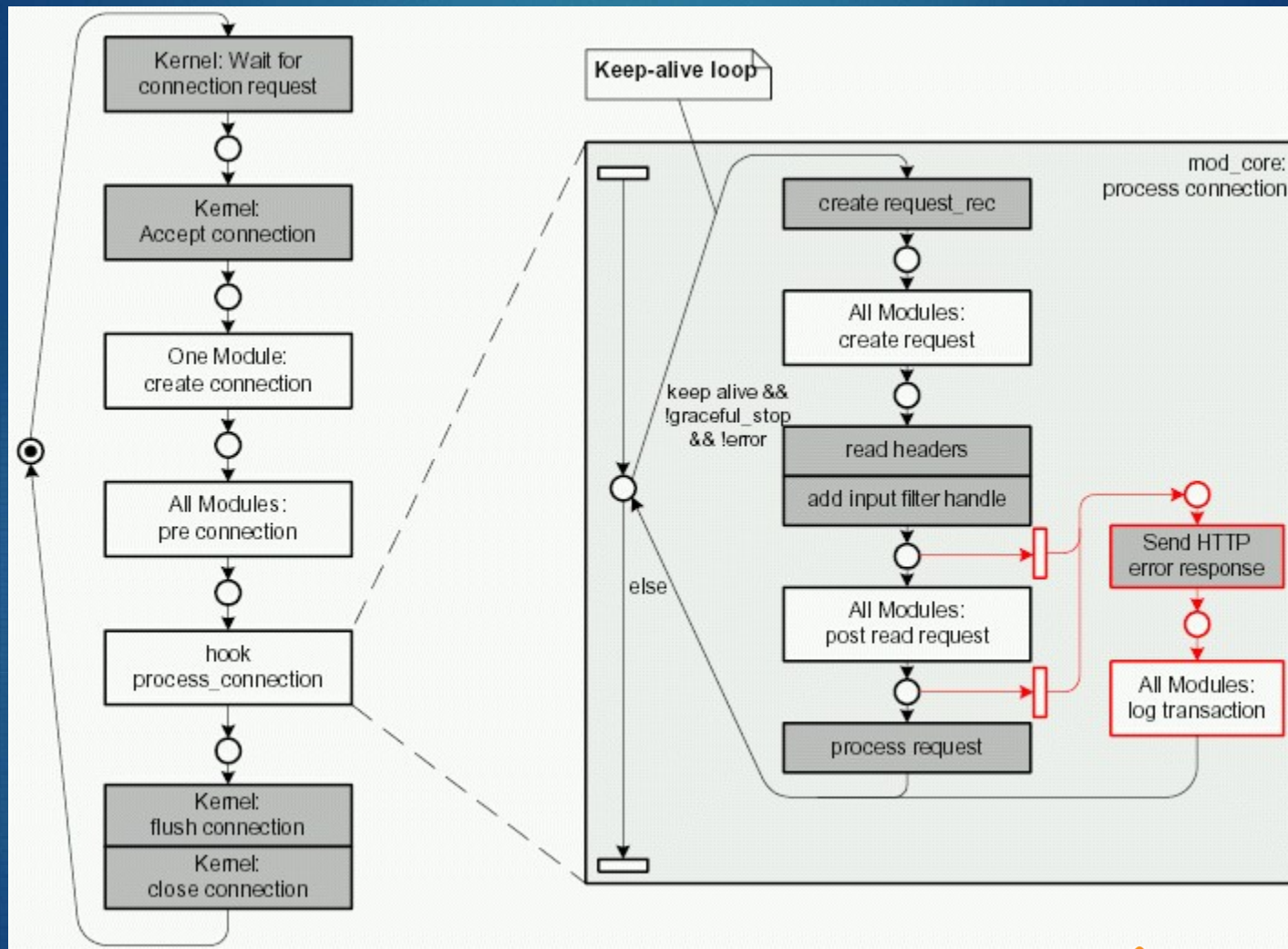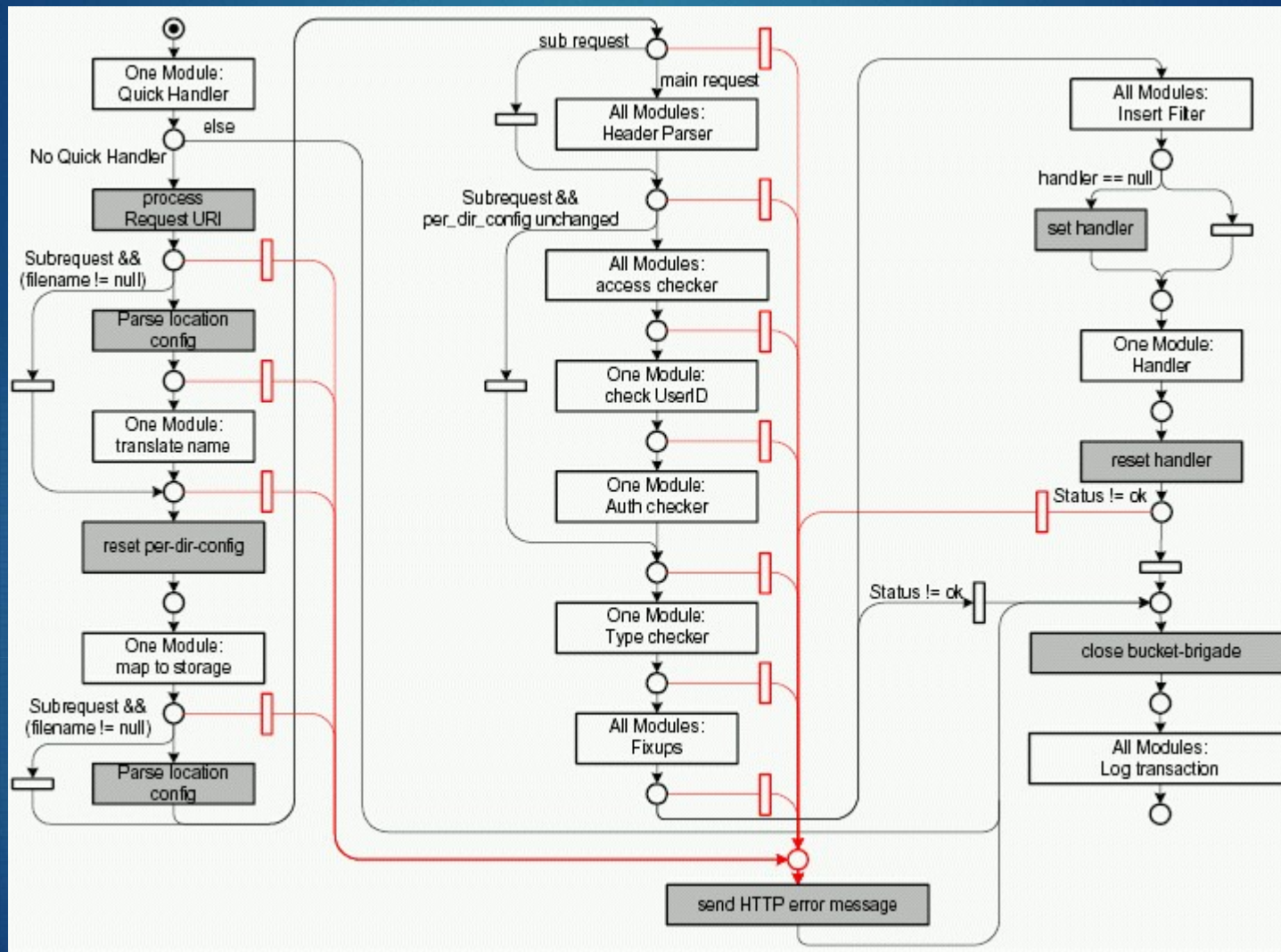
Modules Interface

```c
static int sample_handler(request_rec *r)
{
    if (strcmp(r->handler, "sample")) {
        return DECLINED;
    }
    r->content_type = "text/html";

    if (!r->header_only)
        ap_rputs("The sample page from mod_sample.c\n", r);
    return OK;
}

static void sample_register_hooks(apr_pool_t *p)
{
    ap_hook_handler(sample_handler, NULL, NULL, APR_HOOK_MIDDLE);
}

/* Dispatch list for API hooks */
module AP_MODULE_DECLARE_DATA sample_module = {
    STANDARD20_MODULE_STUFF,
    NULL,              /* create per-dir    config structures */
    NULL,              /* merge  per-dir    config structures */
    NULL,              /* create per-server config structures */
    NULL,              /* merge  per-server config structures */
    NULL,              /* table of config file commands       */
    sample_register_hooks  /* register hooks                  */
};
```

Too many ugly hacks!

/server/main.c:

```
/* This is a hack until we finish the code so that it only reads
    * the config file once and just operates on the tree already in
    * memory.  rbb
    */
```

/server/config.c:

```
char *tmp = ap_malloc(strlen(m->name));
/* FIXME: memory leak, albeit a small one */
```

Erfan Sharafzadeh – Software Architecture Course Presentation

APACHE
HTTP SERVER PROJECT