ESS Outline

- ESS Characteristics
- Requirements
- Security Considerations
- ESS Stack / ESS 2.0
- Instances & Installation
- Linux Care

ESS Outline (cont.)

- Live demo:
  - Staff registration configuration
  - “Personnel Info” screen customization
  - Employee Groups
  - Absence Tracking configuration
  - District roll-out

Characteristics of ESS

- Web based
  - no PC/client software to install/maintain
  - Internet Explorer, Firefox, Chrome
- Integrated with ***QSS***/OASIS
  - Authentication
  - "Live" employee data
  - Configuration data
  - Export files

Characteristics of ESS (cont.)

- Easy, secure (HTTPS) access to employees' data from:
  - Desktop, kiosk, home
- Context-aware help
- Easy to customize and configure
  - Config via GUI, not separate files
  - Changes take effect immediately – no need to re-login

Characteristics of ESS (cont.)

- Ability for certain users to see other employees' data:
  - "Group" supervisor
  - Right granted to user to see employees within own district
  - Right granted to user to see employees across all districts

Characteristics of ESS (cont.)

- QSS Support
  - 4 hours of ESS training bundled with purchase
  - We install ESS and Minisoft ODBC
  - Assist with security, AD and misc config
  - Quick problem resolution
  - Online, up-to-date documentation

Online ESS Manual

http://www.qss.com/webdocs/QSSManuals/getting_started_with_ess.pdf

- Unlike other QSS online doc, URL above is permanent and can be book-marked
- 80+ pages, many screen-shots, regularly updated

ESS Requirements

- Client browser – IE 8+, FF, Chrome
- Linux server – RedHat, SUSE
  - Typically provisioned to become the version-L server
- MS SQLServer or PostgreSQL database
- Minisoft ODBC driver for Version H
- QSS access to Linux and db servers

Roles and Rights

- Rights assigned to role for Read-only, Write/Read, or eXclude access
- Roles matching current app & district are evaluated in pri order (when user has multiple roles)
- If multiple roles specify the same right, the highest pri role determines the right
- Lower pri roles define rights when higher pri roles are neutral (blank)

Roles, Apps, Districts

- Roles grant access to one or more apps and districts
- Only roles granting access to the current app & district are evaluated
- Cannot login unless your role grants access to ESS and to your district

Multi-District Security

- admin -> Security -> Districts: specifies districts available for any QSS web app
- admin -> Customize -> ESS Access: specifies districts specifically for ESS
- admin -> Security -> Roles: to add a new district to an existing roles

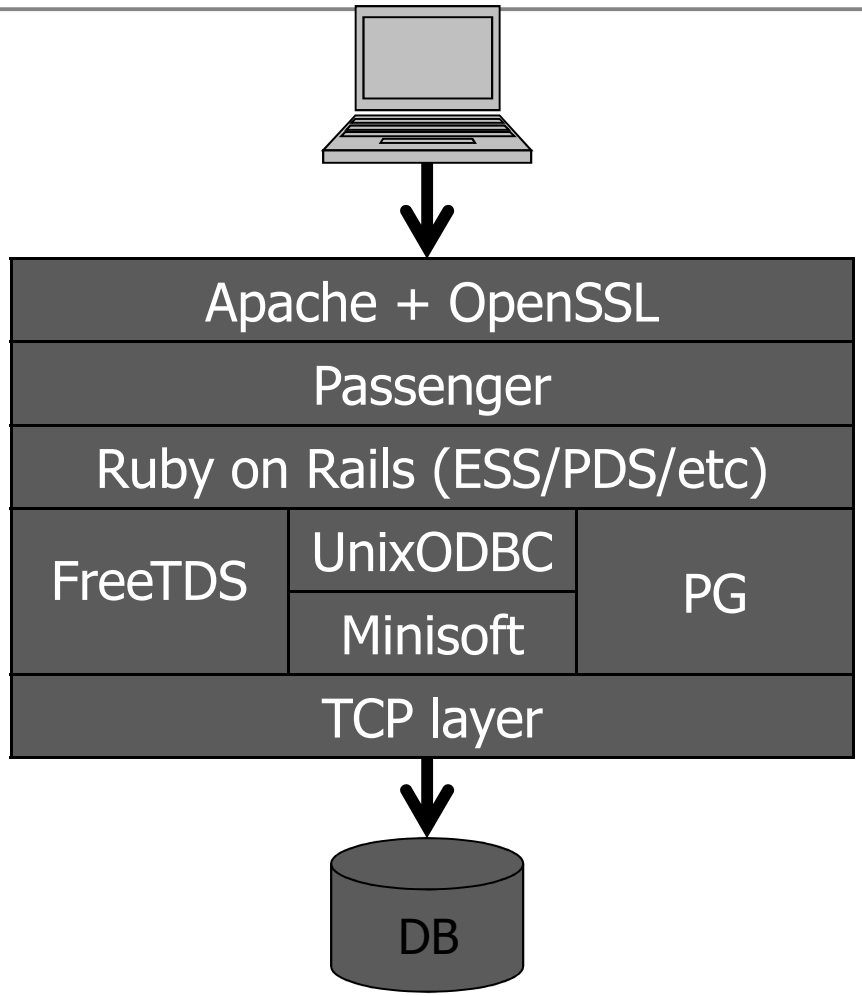
Multi-District Security (cont)

- admin -> Configuration -> Summary: to clone current configuration to target district
- Main district drop-down (top-right) shows districts enabled by ESS and available based on the user's role(s)

ESS Stack

- Self-sufficient -- install all middleware needed by ESS including:
  - Ruby/Rails source
  - Apache source
  - openSSL source
  - unixODBC source
  - FreeTDS source
- Compile all on customer server

Web App Software Stack



ESS 2.0

- Many internal changes moving to Rails 3.2.8
- Some minor enhancements, bug fixes
- Possibly better performance (TBD)

ESS Install/Update Overview

- Upload tarball to customer server
- Extract and run install script
- Build software stack in staging area
- Activate staging area
- Apply database structural changes
- Manually update config files if needed
- Manually update hp3k schema if needed

ESS Instances

- ESS may be installed in multiple "instances" (environments)
- Each instance is completely separate
- Typically "test", "training", "qa" etc…
- Usually configured for separate databases

ESS File Layout

- /opt/qss[/*instance*]/hrsweb/ess/ –
  ESS source (no customer-modifiable files)
- /etc/opt/qss[/*instance*]/hrsweb/–
  configs (customer-modifiable)
- /var/opt/qss[/*instance*]/hrsweb/ess/ –
  log files (grow without bound)
- The "/*instance*" portion is omitted for the default "production" instance

File Layout
/opt/qss[/*instance*]/hrsweb/

- hrsweb (core), rails (core) and app-specific time-stamped subdirectories
- The *rails* subdir contains the compiled software stack (i.e. Apache, Ruby, etc)
- Symlinks point to most recent subdirs
- Most recent installation log files

User-Editable Config Files

/etc/opt/qss[/*instance*]/hrsweb

- common.sysconfig – hostname & IP addr
- environment_init.rb – email params
- proxy_*.conf – Apache configuration
  - proxy_common.conf – SSL X.509 key/cert
- public subdir – logo & home page content gets uploaded here

Log Files

/var/opt/qss[/*instance*]/hrsweb/

- Separate subdirs for each web app
  - /var/opt/qss/hrsweb/ess/ – ESS "production" log files
  - /var/opt/qss/*test*/hrsweb/ess/ – ESS "test" instance log files
- Files grow without bound!

Log Files (cont)

- Apache log files:
  - access.log
  - error.log (Ruby $stderr.puts goes here too)
  - rewrite.log
- Ruby on Rails ESS log files
  - …/ess/production.log
  - …/ess/development.log (debugging only)

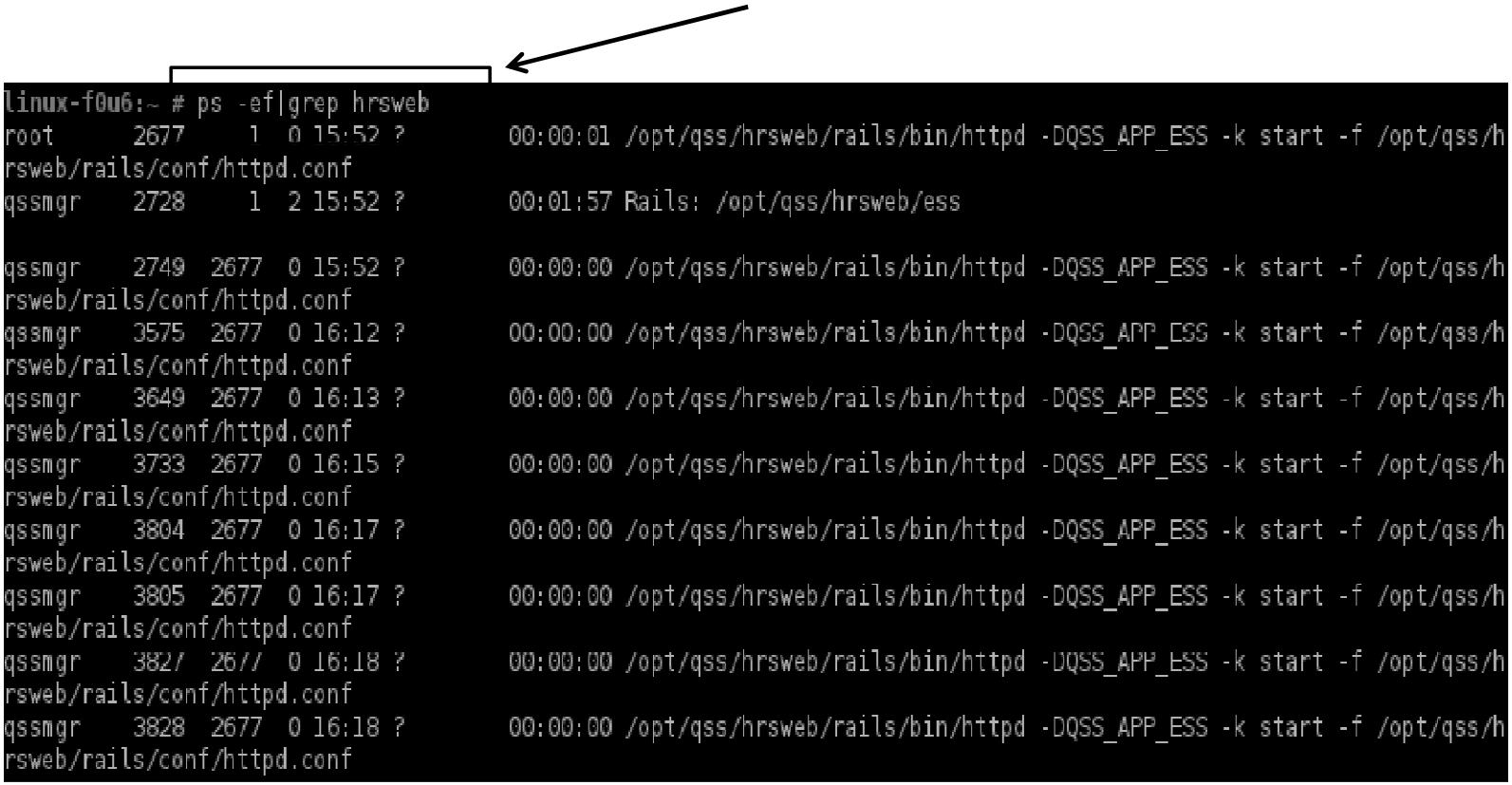
Network Considerations

- Listen to all server IP addresses vs. one server IP address?
- Use IP address aliasing (multiple addresses per NIC)?
- Which TCP ports?
  ESS default (29295) or HTTPS default (443)?
- Multiple IP addresses are required for multiple HTTPS apps using the same ports on the same machine

Startup/Shutdown

- Automatically integrated into server startup/shutdown sequence via the Linux *chkconfig* command
- /etc/init.d/qss_hrsweb_ctl [-i inst1,inst2…] [-a app1,app2…] {start | stop | restart}, e.g.:
  - **/etc/init.d/qss_hrsweb_ctl –a ess start**
- Default is all instances, all applications

Is ESS Running?



Linux Care

- Must have OS software update subscription
- Should stay current on OS patches
- Should examine OS log files periodically
- Should monitor disk space utilization