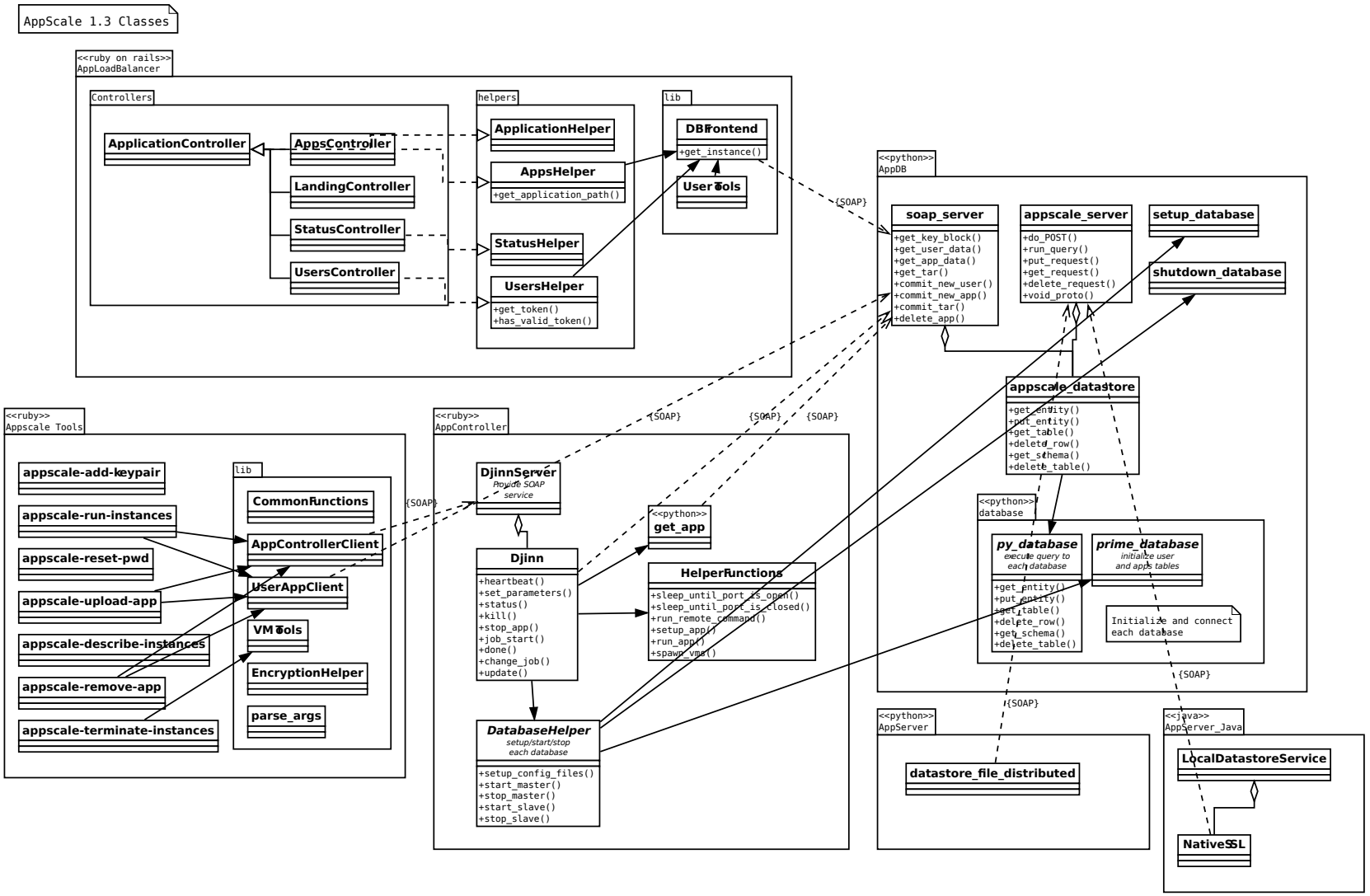
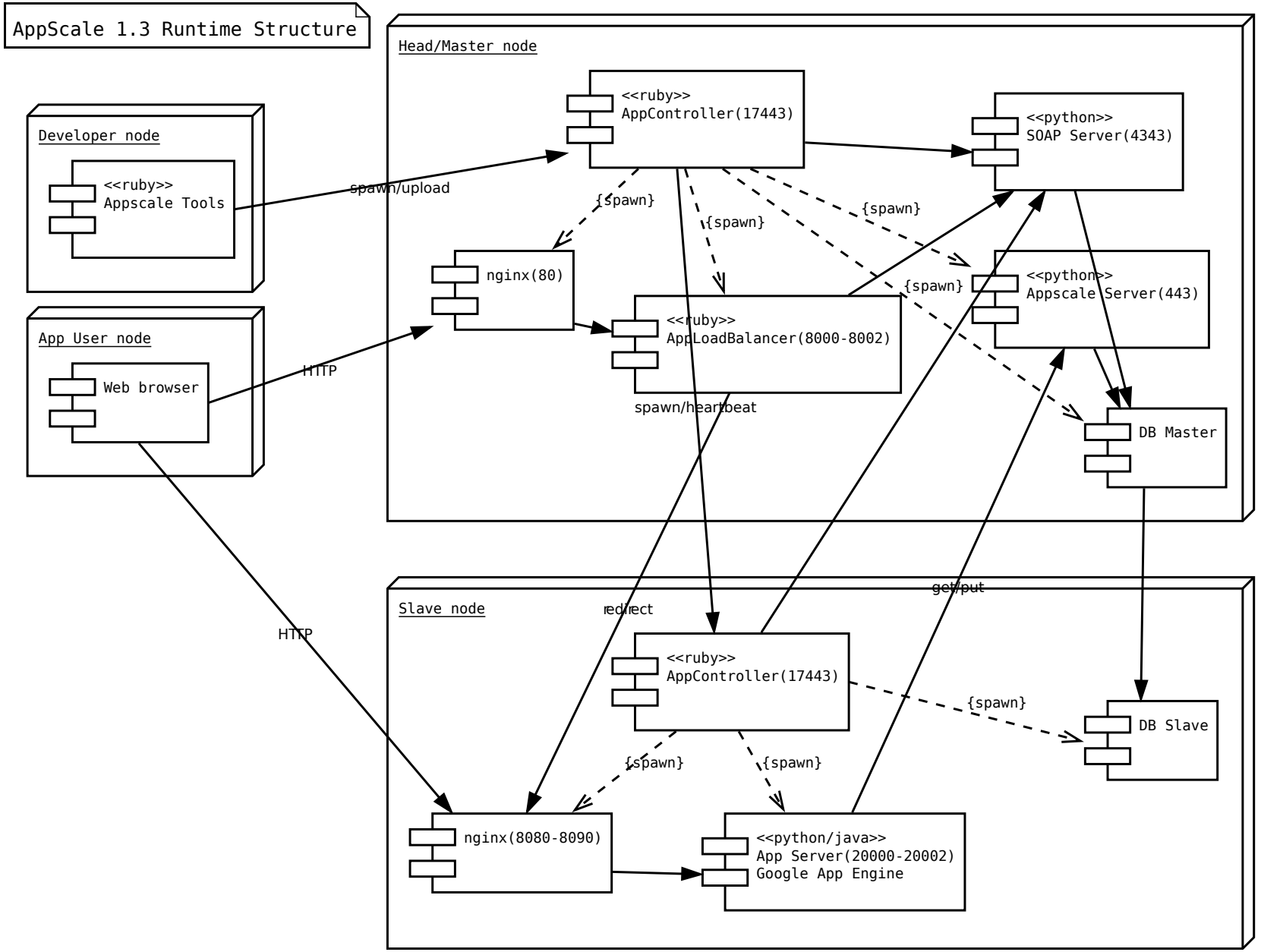
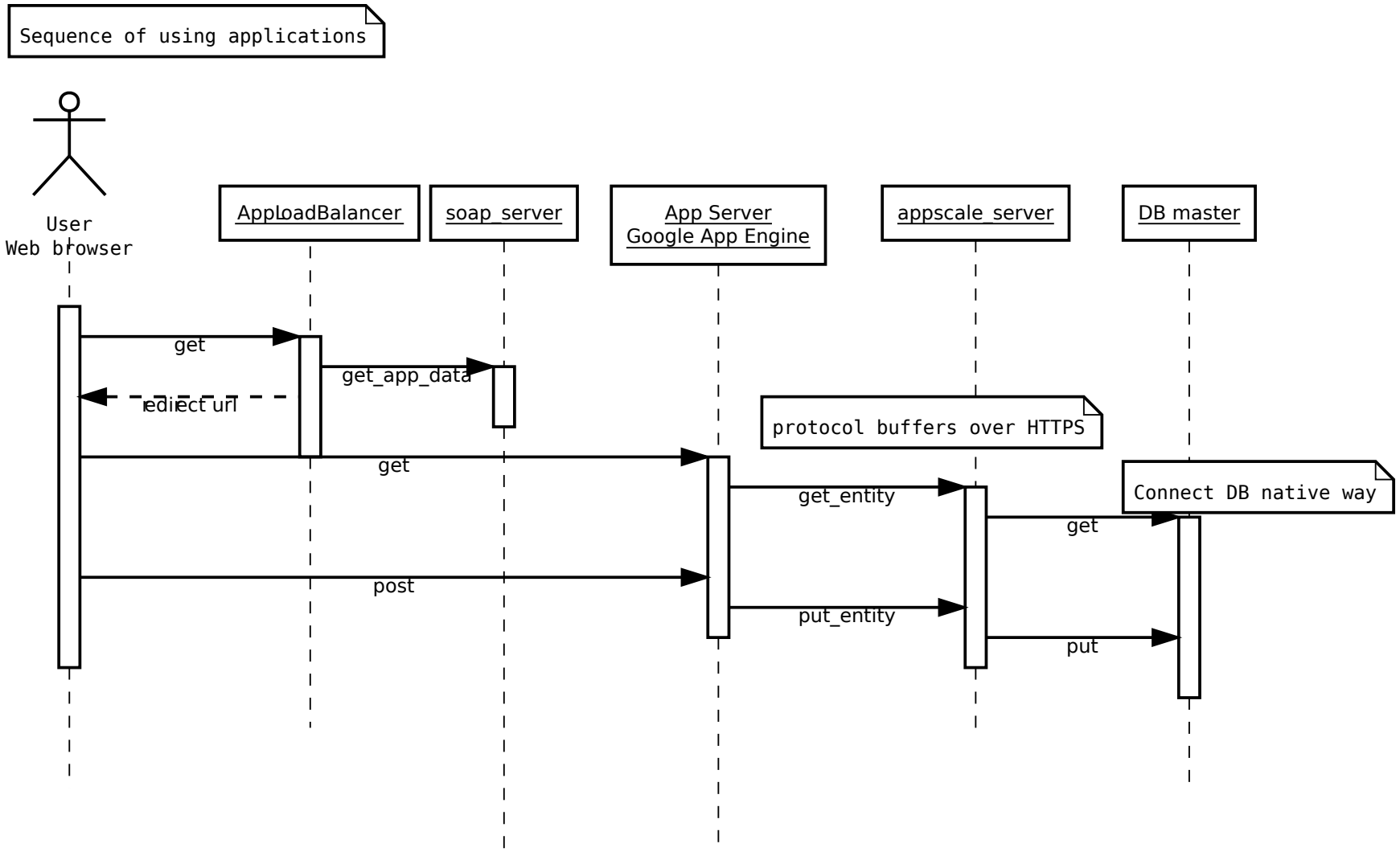
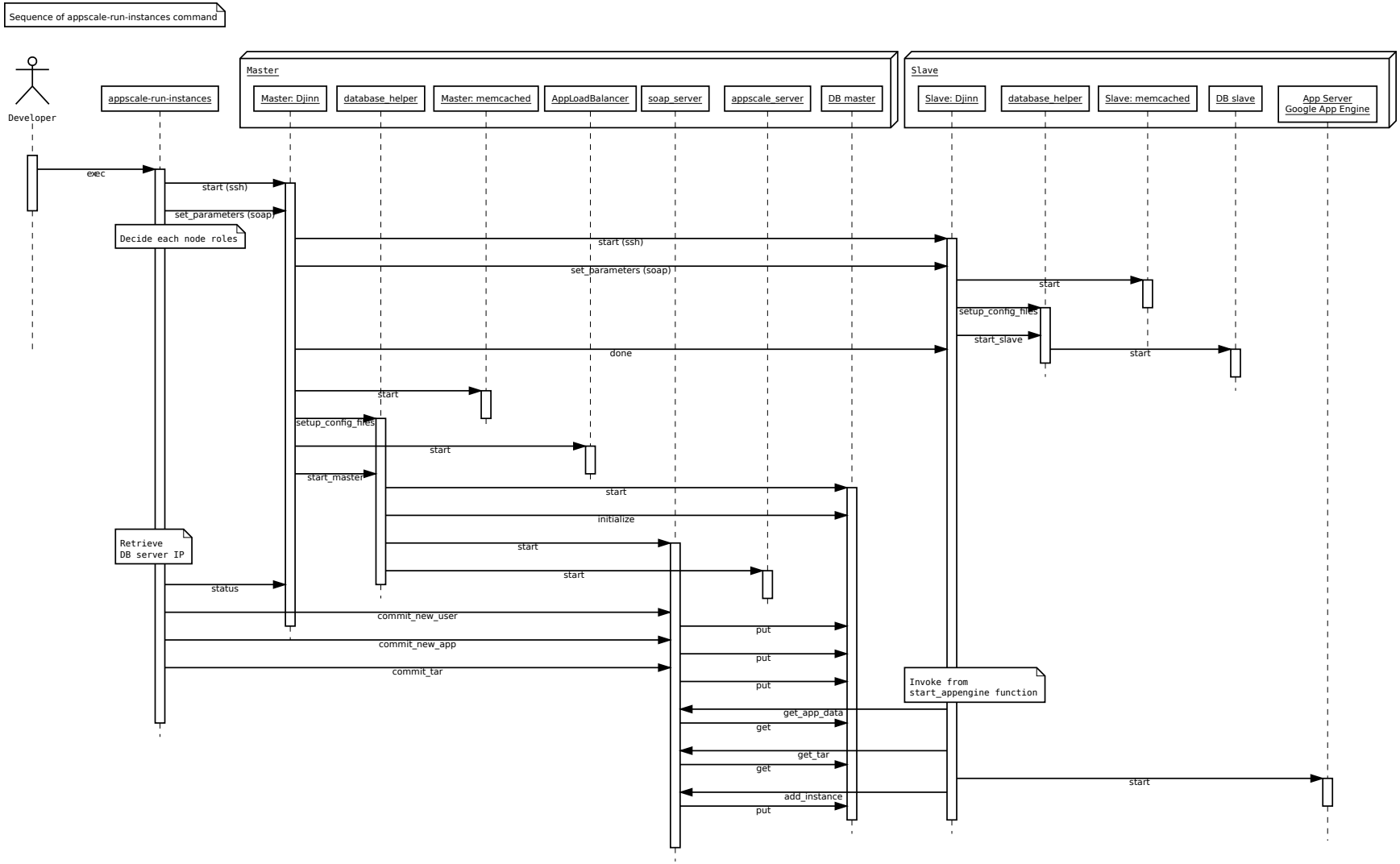
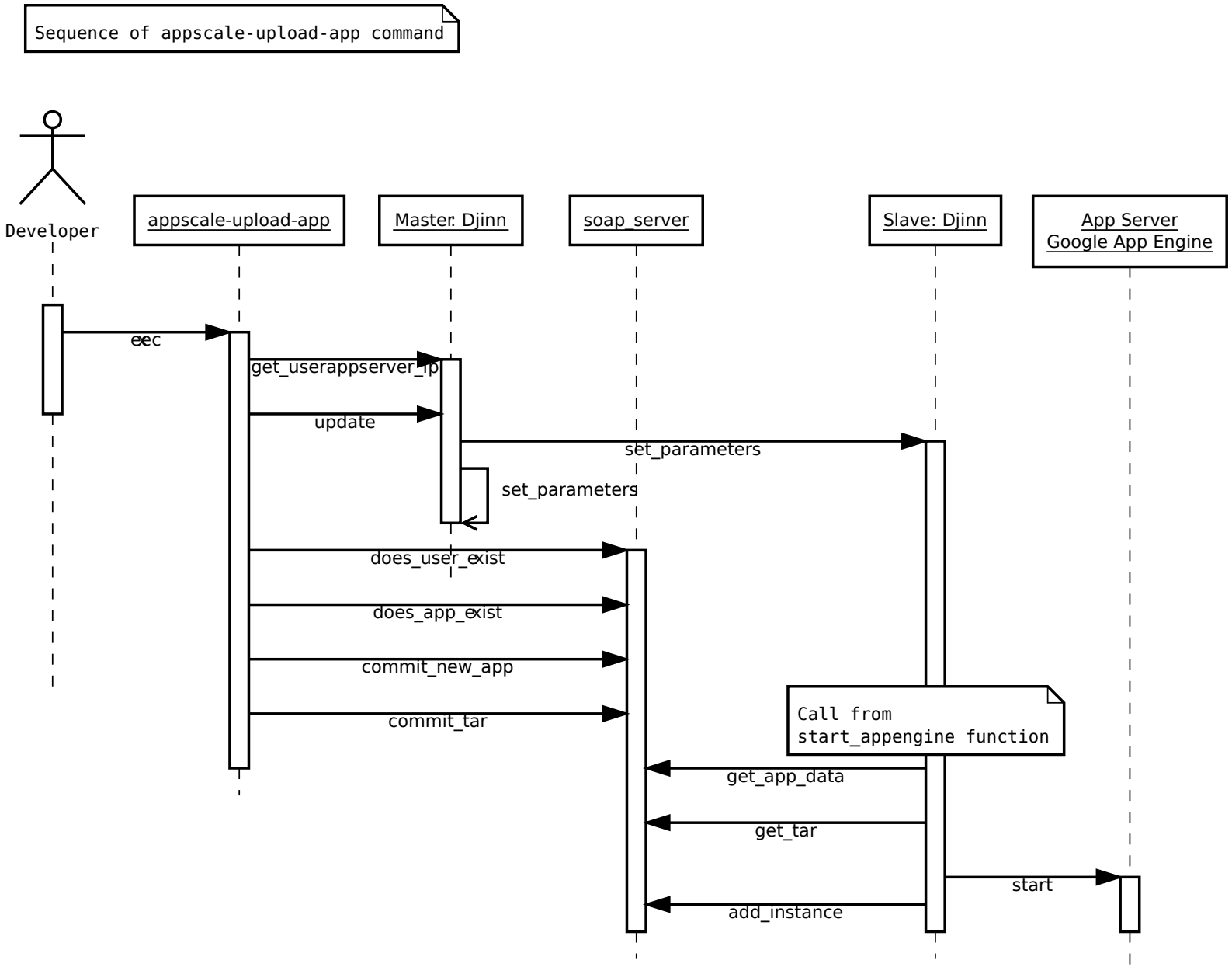
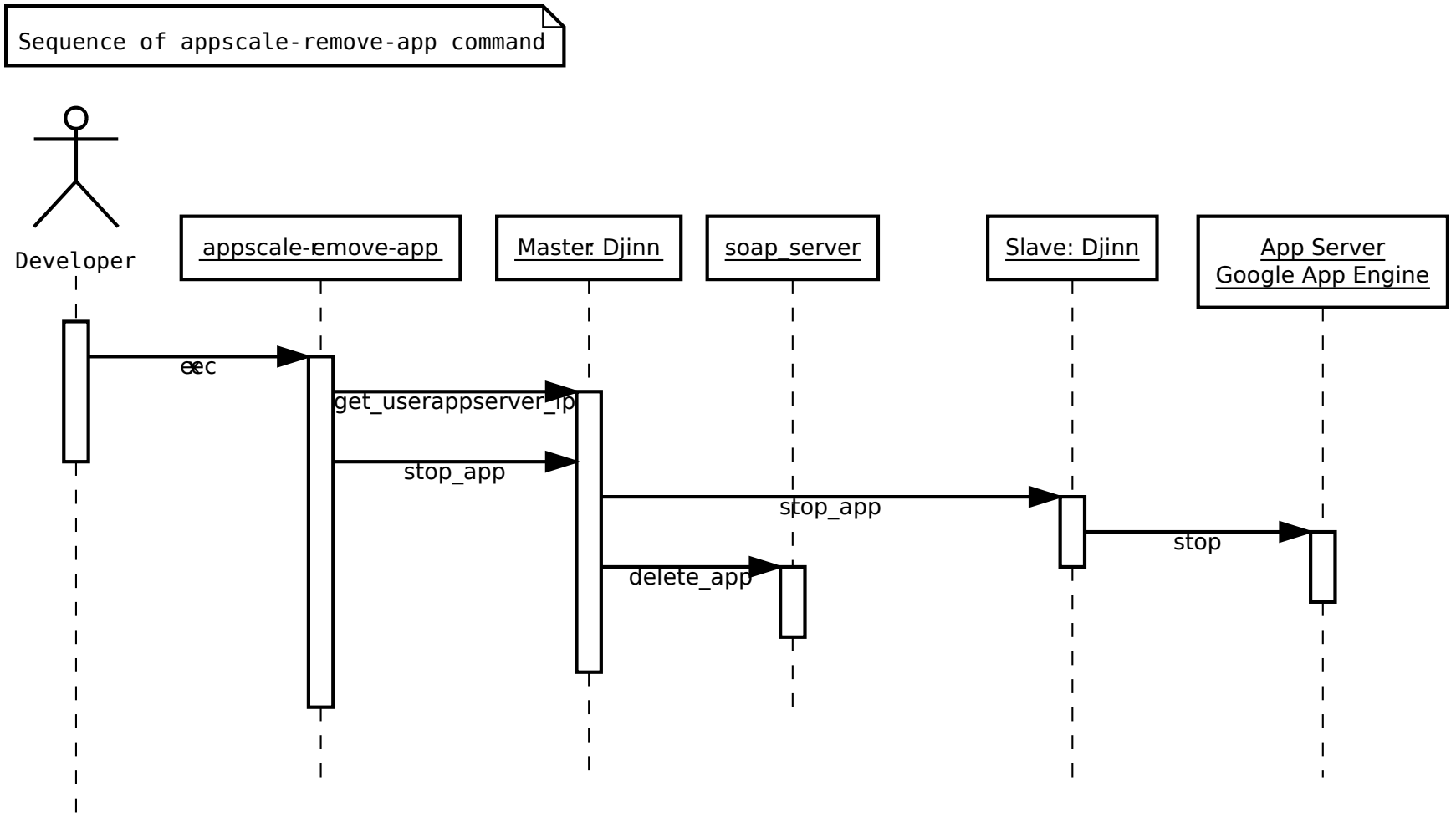

AppScale 1.3 Classes
<<ruby on rails>>
AppLoadBalancer
Controllers
ApplicationController
AppsController
LandingController
StatusController
UsersController
helpers
ApplicationHelper
AppsHelper
+get_application_path()
StatusHelper
UsersHelper
+get_token()
+has_valid_token()
lib
DBFrontend
+get_instance()
UserTools
{SOAP}
<<python>>
AppDB
soap_server
+get_key_block()
+get_user_data()
+get_app_data()
+get_tar()
+commit_new_user()
+commit_new_app()
+commit_tar()
+delete_app()
appscale_server
+do_POST()
+run_query()
+put_request()
+get_request()
+delete_request()
+void_proto()
setup_database
shutdown_database
appscale_datastore
+get_entity()
+put_entity()
+get_table()
+delete_row()
+get_schema()
+delete_table()
<<python>>
database
py_database
execute query to each database
prime_database
initialize user and apps tables
+get_entity()
+put_entity()
+get_table()
+delete_row()
+get_schema()
+delete_table()
Initialize and connect each database
<<ruby>>
Appscale Tools
appscale-add-keypair
appscale-run-instances
appscale-reset-pwd
appscale-upload-app
appscale-describe-instances
appscale-remove-app
appscale-terminate-instances
lib
CommonFunctions
AppControllerClient
UserAppClient
VMTools
EncryptionHelper
parse_args
{SOAP}
<<ruby>>
AppController
DjinnServer
Provide SOAP service
Djinn
+heartbeat()
+set_parameters()
+status()
+kill()
+stop_app()
+job_start()
+done()
+change_job()
+update()
<<python>>
get_app
HelperFunctions
+sleep_until_port_is_open()
+sleep_until_port_is_closed()
+run_remote_command()
+setup_app()
+run_app()
+spawn_vm()
DatabaseHelper
setup/start/stop each database
+setup_config_files()
+start_master()
+stop_master()
+start_slave()
+stop_slave()
<<python>>
AppServer
datastore_file_distributed
<<java>>
AppServer_Java
LocalDatastoreService
NativeSSL


AppScale 1.3 Runtime Structure
Head/Master node
<<ruby>>
AppController(17443)
<<python>>
SOAP Server(4343)
Developer node
<<ruby>>
Appscale Tools
spawn/upload
{spawn}
{spawn}
{spawn}
{spawn}
nginx(80)
<<python>>
Appscale Server(443)
App User node
Web browser
HTTP
<<ruby>>
AppLoadBalancer(8000-8002)
spawn/heartbeat
DB Master
Slave node
redirect
get/put
HTTP
<<ruby>>
AppController(17443)
{spawn}
DB Slave
{spawn}
{spawn}
nginx(8080-8090)
<<python/java>>
App Server(20000-20002)
Google App Engine


Sequence of using applications
User
Web browser
AppLoadBalancer
soap_server
App Server
Google App Engine
appscale_server
DB master
get
get_app_data
redirect url
protocol buffers over HTTPS
get
get_entity
Connect DB native way
get
post
put_entity
put


Sequence of appscale-run-instances command
Developer
appscale-run-instances
Master
Master: Djinn
database_helper
Master: memcached
AppLoadBalancer
soap_server
appscale_server
DB master
Slave
Slave: Djinn
database_helper
Slave: memcached
DB slave
App Server
Google App Engine
exec
start (ssh)
set_parameters (soap)
Decide each node roles
start (ssh)
set_parameters (soap)
start
setup_config_files
start_slave
start
done
start
setup_config_files
start
start_master
start
initialize
start
start
Retrieve
DB server IP
status
commit_new_user
put
commit_new_app
put
commit_tar
put
Invoke from
start_appengine function
get_app_data
get
get_tar
get
start
add_instance
put


Sequence of appscale-upload-app command
Developer
appscale-upload-app
Master: Djinn
soap_server
Slave: Djinn
App Server
Google App Engine
exec
get_userappserver_ip
update
set_parameters
set_parameters
does_user_exist
does_app_exist
commit_new_app
commit_tar
Call from
start_appengine function
get_app_data
get_tar
start
add_instance


Sequence of appscale-remove-app command
Developer
appscale-remove-app
Master: Djinn
soap_server
Slave: Djinn
App Server
Google App Engine
exec
get_userappserver_ip
stop_app
stop_app
stop
delete_app