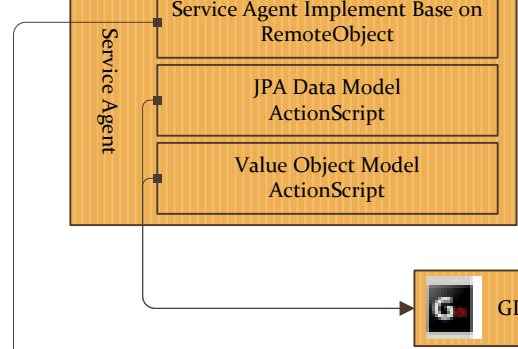
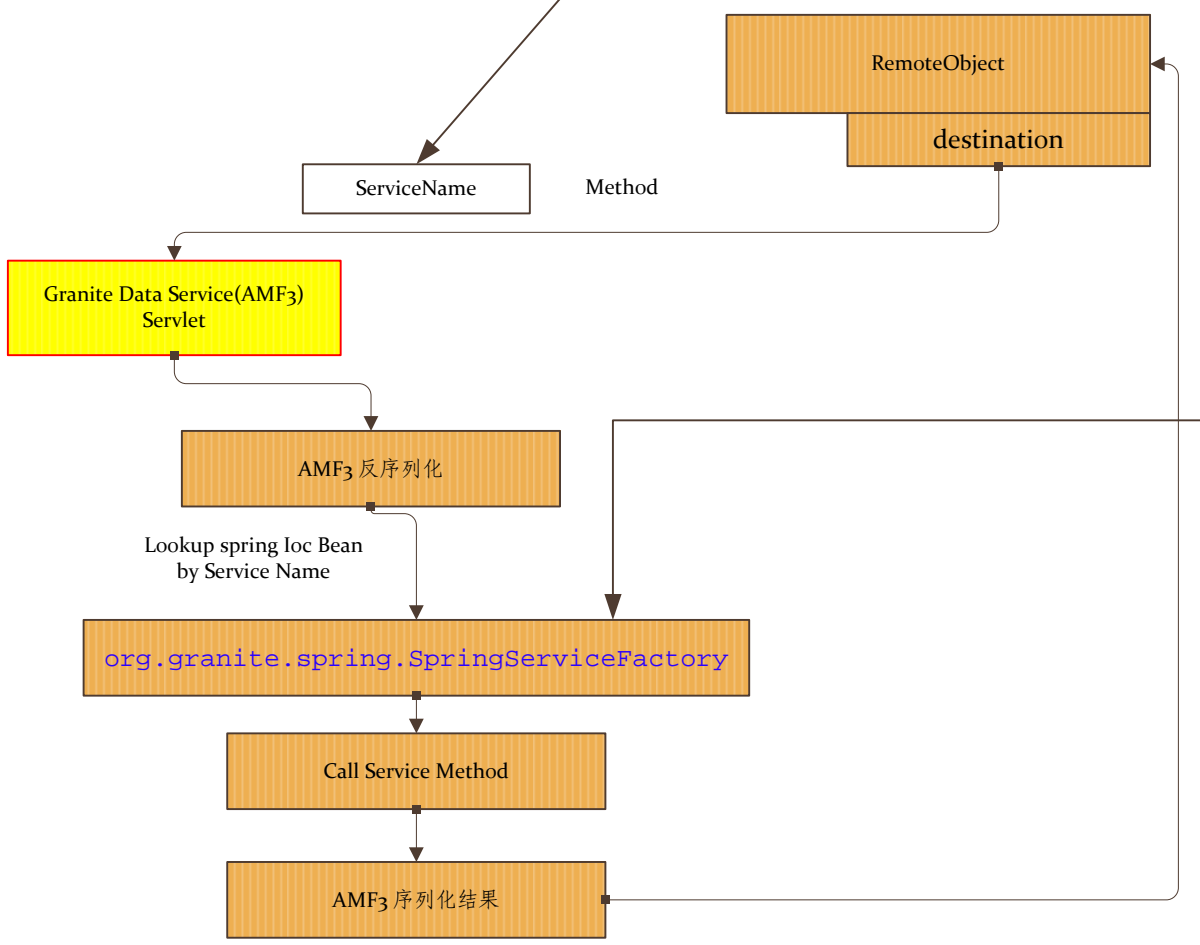
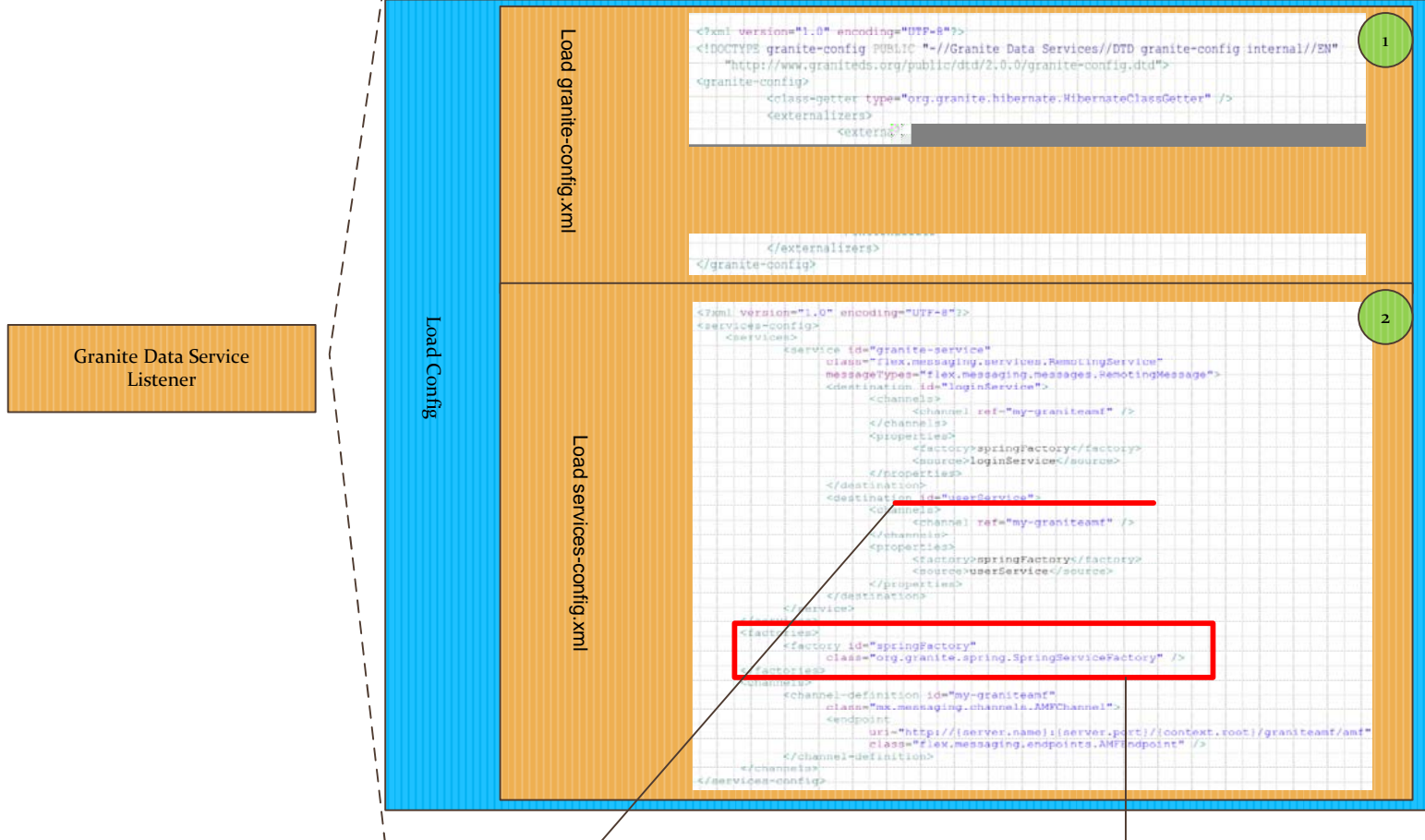
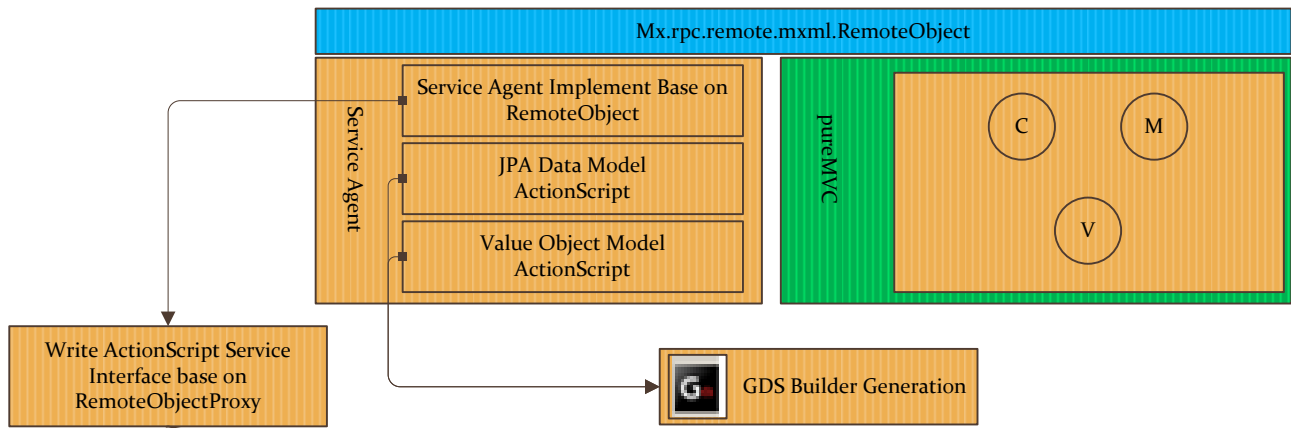


Write JAVA Service Interface base on RemoteObjectProxy







```

package com.csgjs.service
{
    import flash.events.EventDispatcher;
    import mx.rpc.AbstractOperation;
    import mx.rpc.AsyncToken;
    import mx.rpc.Responder;
    import mx.rpc.events.FaultEvent;
    import mx.rpc.events.ResultEvent;
    import mx.rpc.remoting.mxml.RemoteObject;
    import org.as3commons.logging.ILogger;
    import org.as3commons.logging.LoggerFactory;
    public class RemoteObjectProxy extends EventDispatcher
    {
        private var logger:ILogger = LoggerFactory.getClassLogger(RemoteObjectProxy);
        public function RemoteObjectProxy(destination:String=null)
        {
            logger.info("RemoteObjectProxy has been created!", this);
            ro= new RemoteObject(destination);
        }
        private var ro:RemoteObject;
        public function get service():RemoteObject{
            return ro;
        }
        public function set service(value:RemoteObject):void{
            logger.info("service property has been changed!", value);
            ro = value;
        }
        protected function getOperation(name:String):AbstractOperation{
            return ro.getOperation(name);
        }
        protected function call(method:String, ...args):IASyncOperation{
            logger.info("call remote "+method+" function start.", args);
            var o:AbstractOperation = getOperation(method);
            o.arguments = args;
            var token:AsyncToken = o.send();
            var result:IASyncOperation = new AbstractAsyncOperation();
            token.addResponder(new Responder(function(e:ResultEvent):void{
                result.dispatchResult(e.result);
            }, function(e:FaultEvent):void{
                result.dispatchError(e.fault);
            }));
            logger.info("call remote "+method+" function end.", args);
            return result;
        }
    }
}

```