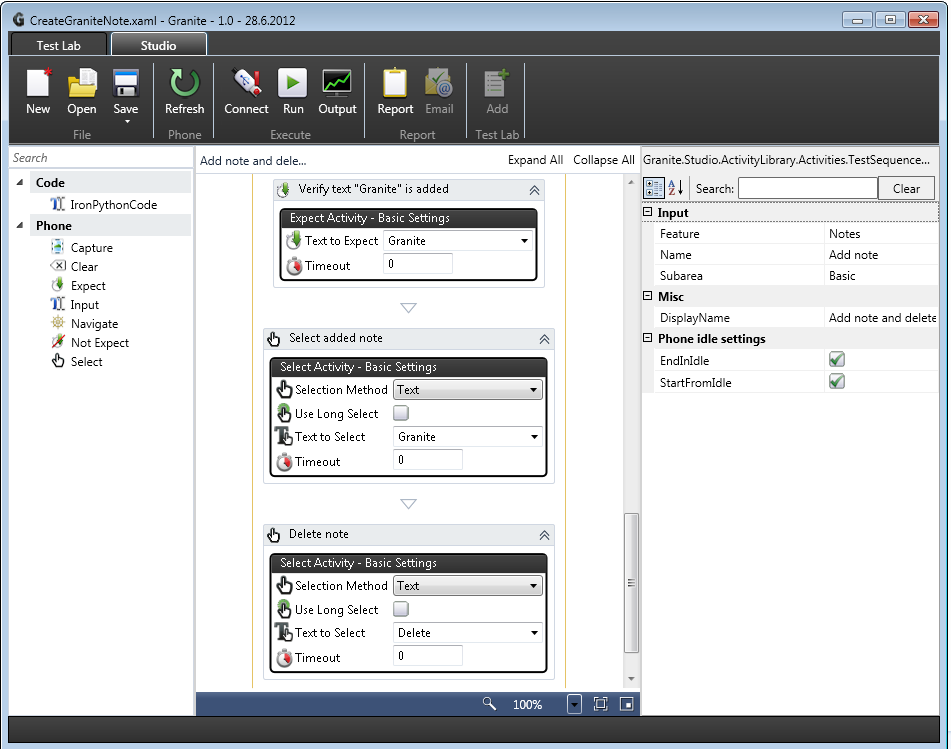
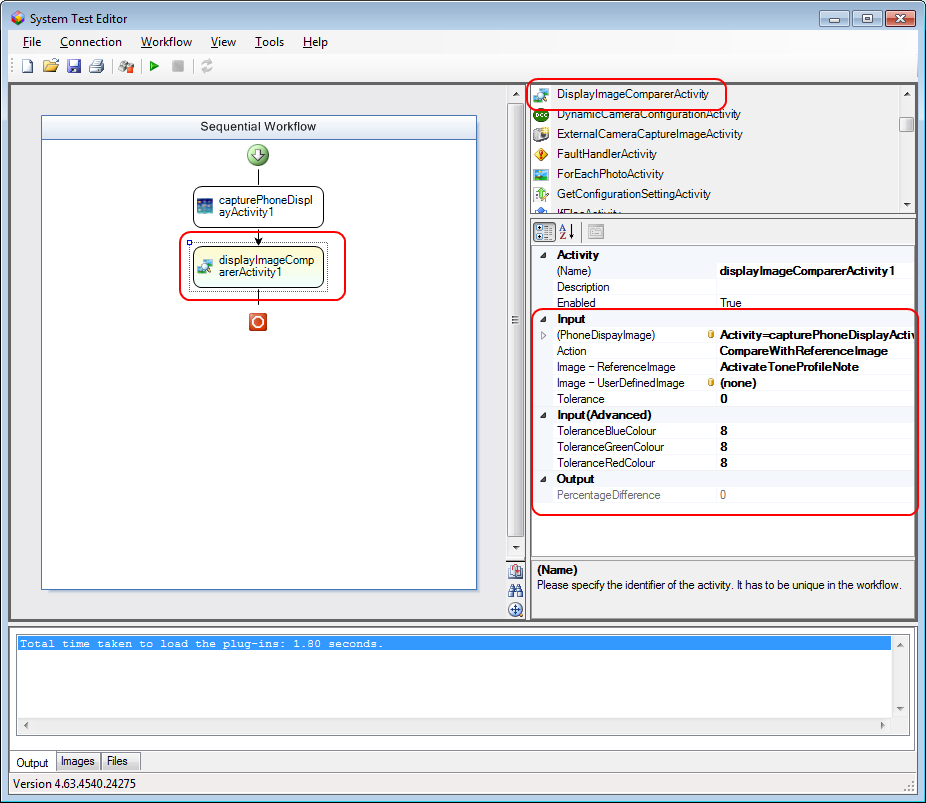
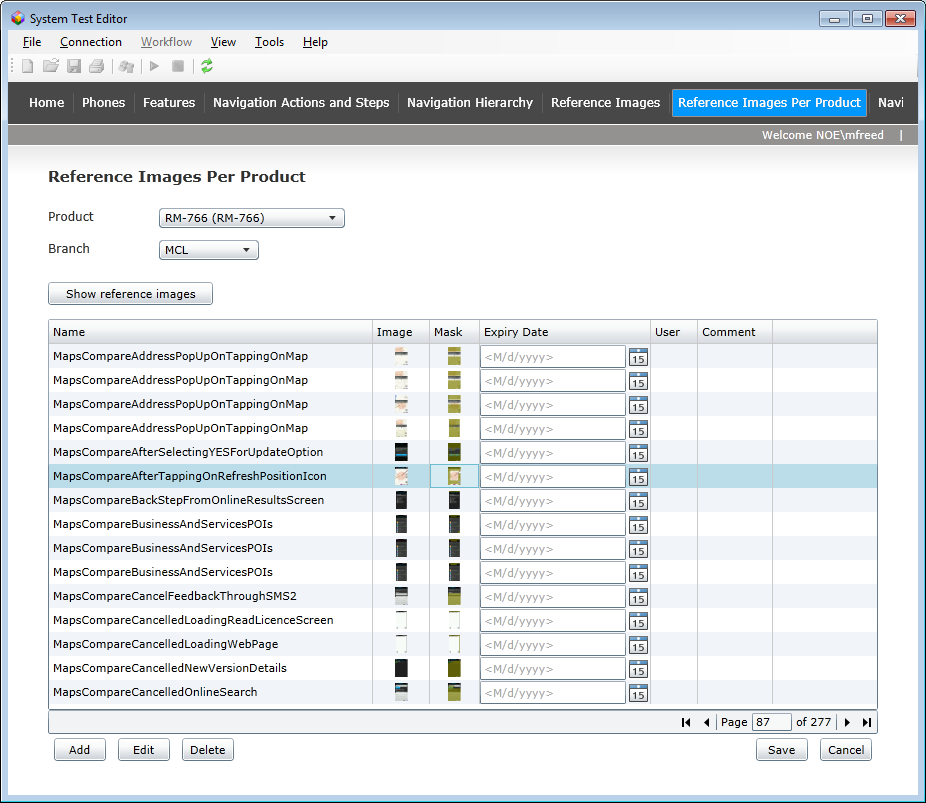
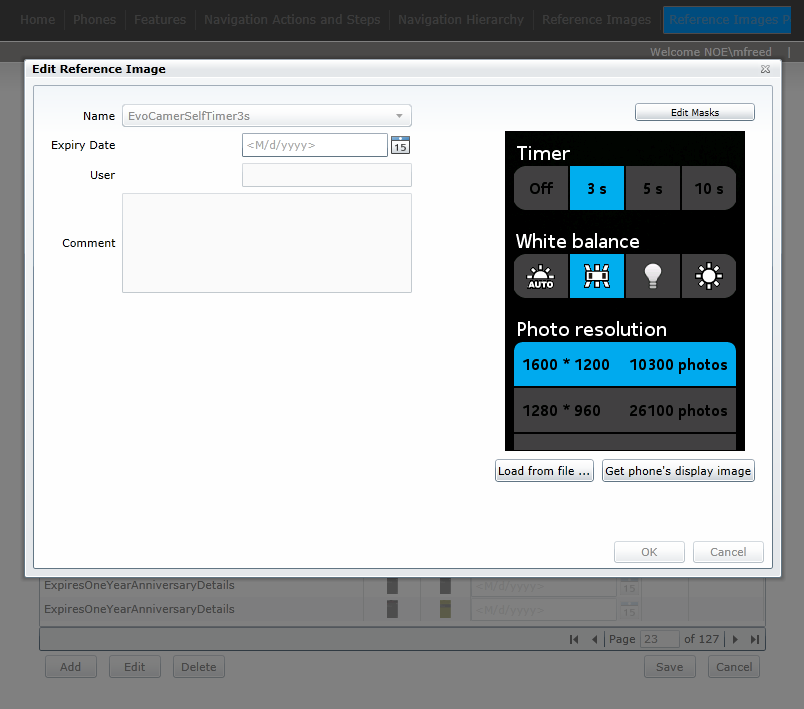
**Overview**

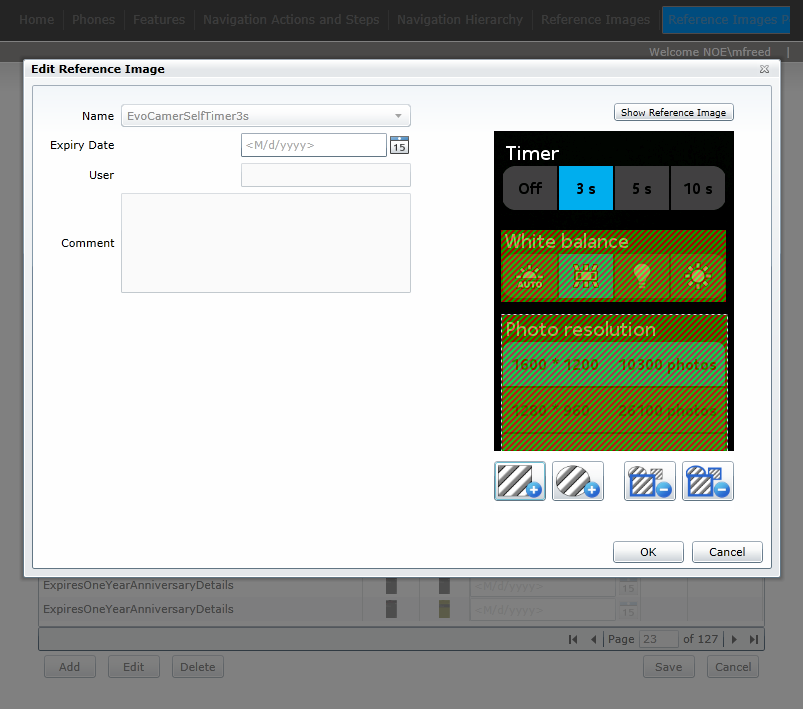


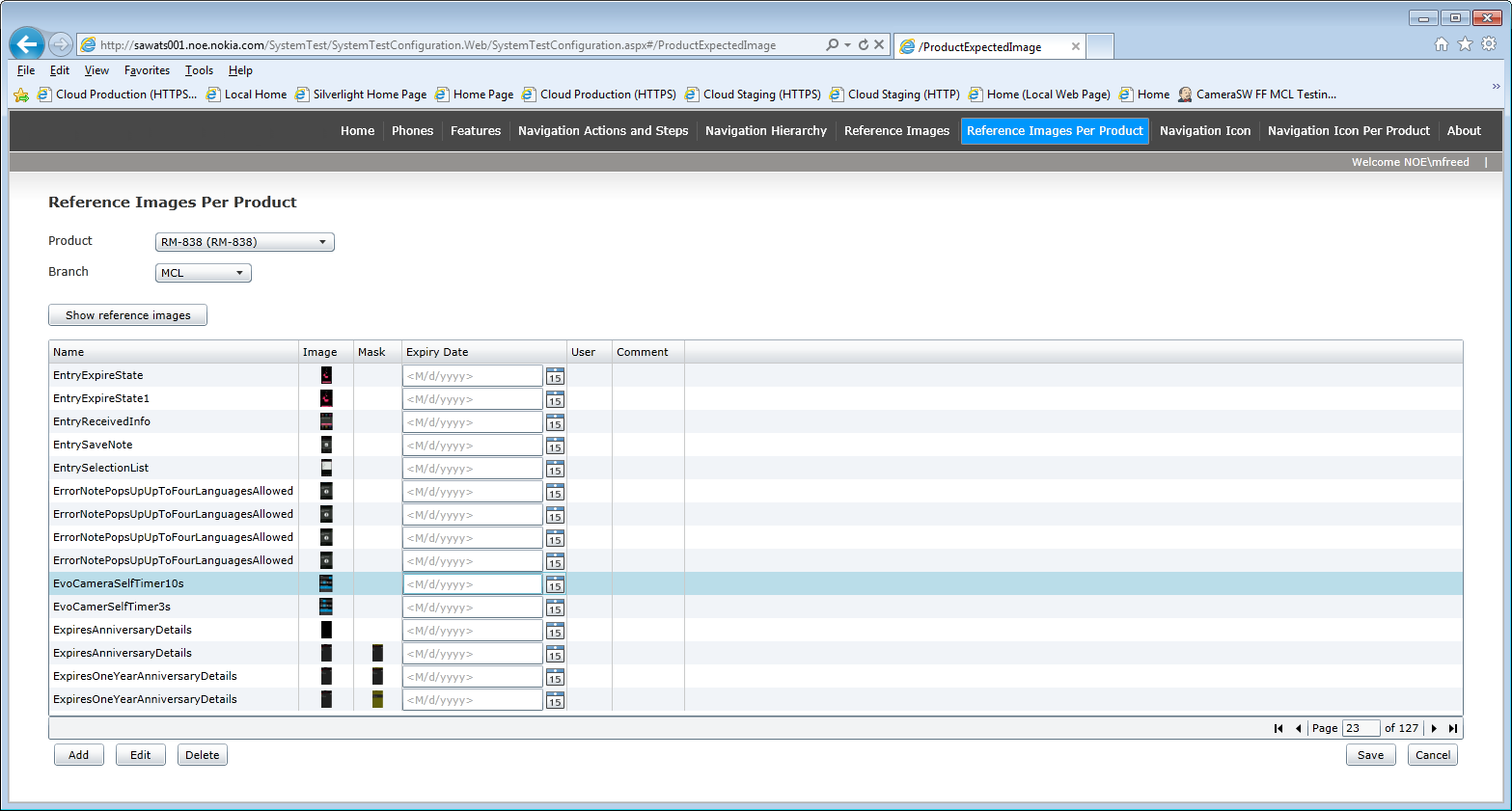


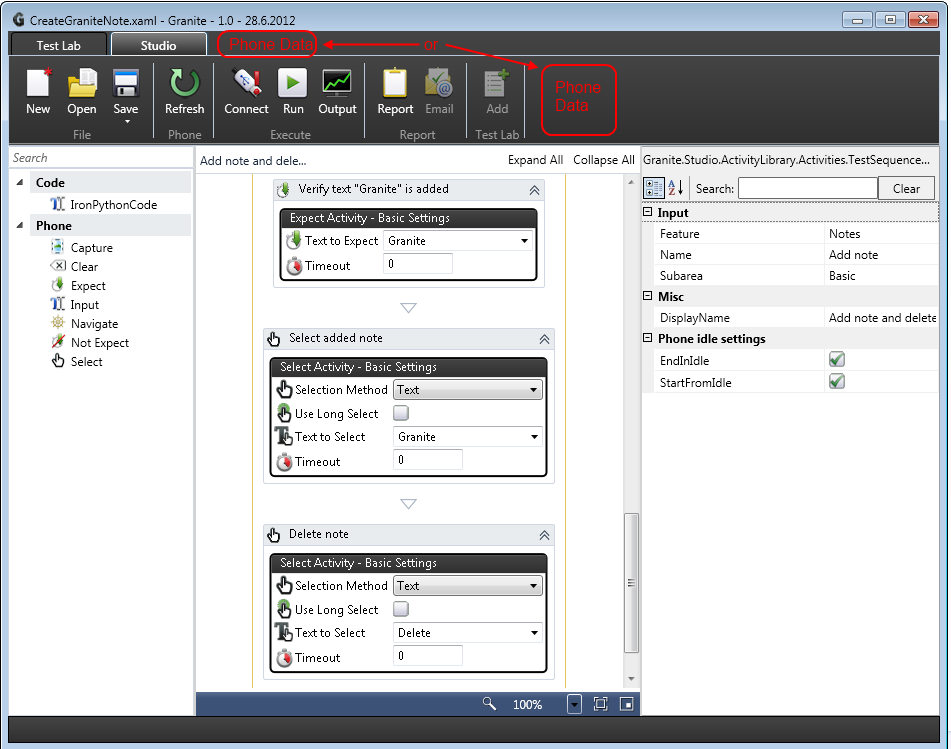


From Granite we need to be able to add multiple reference images and masks:







*Note: This is only initial draft and may be changed*

**Key Decisions**

1. Working as one single team (with single source) vs. Separate teams with plug-in architecture?  
   1. We can either move all necessary source STE source code into Granite project (and convert names/namespaces)  
        
      May require closer co-ordination and communication geographically.  
      Adherence to Granite colour schemes, UI look & feel, architecture.  
      Converting all source code / namespaces and fitting it in with in Granite solution  
      Cutting out and removing code that is not to be ported across.  
        
      or
   2. Provide generic plug-in architecture  
        
      Allow any groups (with the necessary skills) to create their own plug-ins.  
      For some groups this could provide a lot freedom / flexibility.  
      (This could also be future functionality that is developed at a later date)

Display image comparison workflow activity is more than just a single DLL. Makes use of the following assemblies:  
Nokia.Sienna.Workflow.Activities.Device.Phone.Image.dll

Nokia.Sienna.Common.dll

Nokia.Sienna.Common.Image.DefectDetection.dll

Nokia.Sienna.Common.UI.Controls.dll

Nokia.Sienna.Device.Phone.dll

Nokia.Sienna.Device.Phone.Data.dll

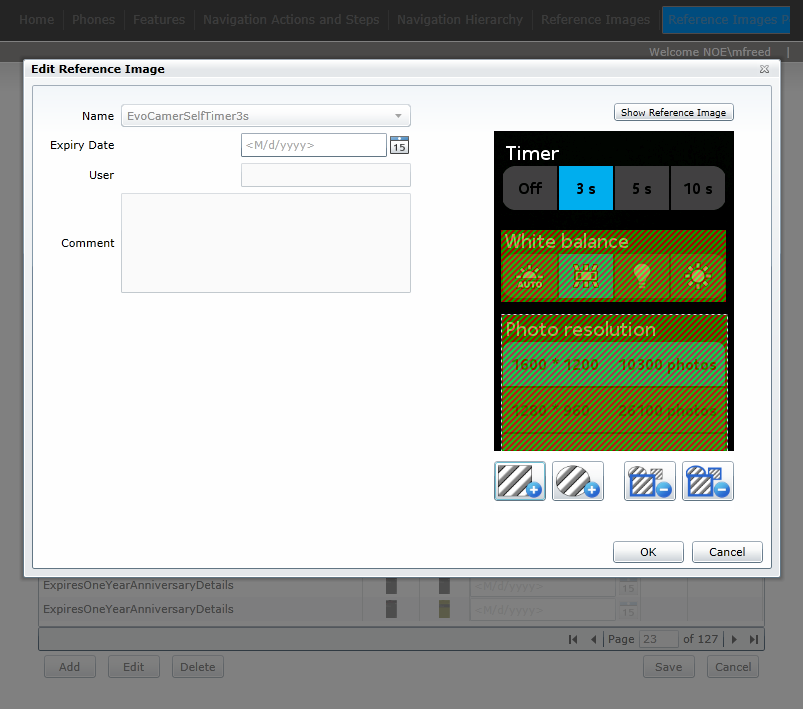
Nokia.Sienna.Device.Phone.Image.DataProvider.dll

Nokia.Sienna.Device.Phone.Product.DataProvider.dll

Nokia.Sienna.SystemTest.Device.Phone.dll

Nokia.Sienna.Workflow.Services.dll

Nokia.Sienna.Device.Lighting.dll

+ SystemTestConfiguration project (web pages)  


Action: Provide Wekey with access to Granite GIT repository on [\\oucifs1\fp\_ts\_fi\work\](file:///\\oucifs1\fp_ts_fi\work\)

**Proposal:**

1. Do quick initial prototype, without converting all code, just creating a Workflow 4.0 PhoneDisplayComparerActivity which uses the same underlying assemblies as those used in STE, just to check there are no technical issues with this activity running inside Granite Framework.
2. Do complete code analysis and identify the code we need to port over to Granite.
3. Port the code over (workflow activity and dependencies).  
   (Wekey)
4. Create new version of SystemTestConfiguration web pages for Granite and cloud database.  
   (Mark)

1. Paying for the cloud?  
     
   Display image comparison requires multiple reference images + masks to be stored per product in the cloud.

Account for SQL Azure databases + web pages.  
Domain name.

1. Technical solution could use existing STE SQL Azure database (short-term only) or a completely new SQL Azure database.
2. Cloud architecture

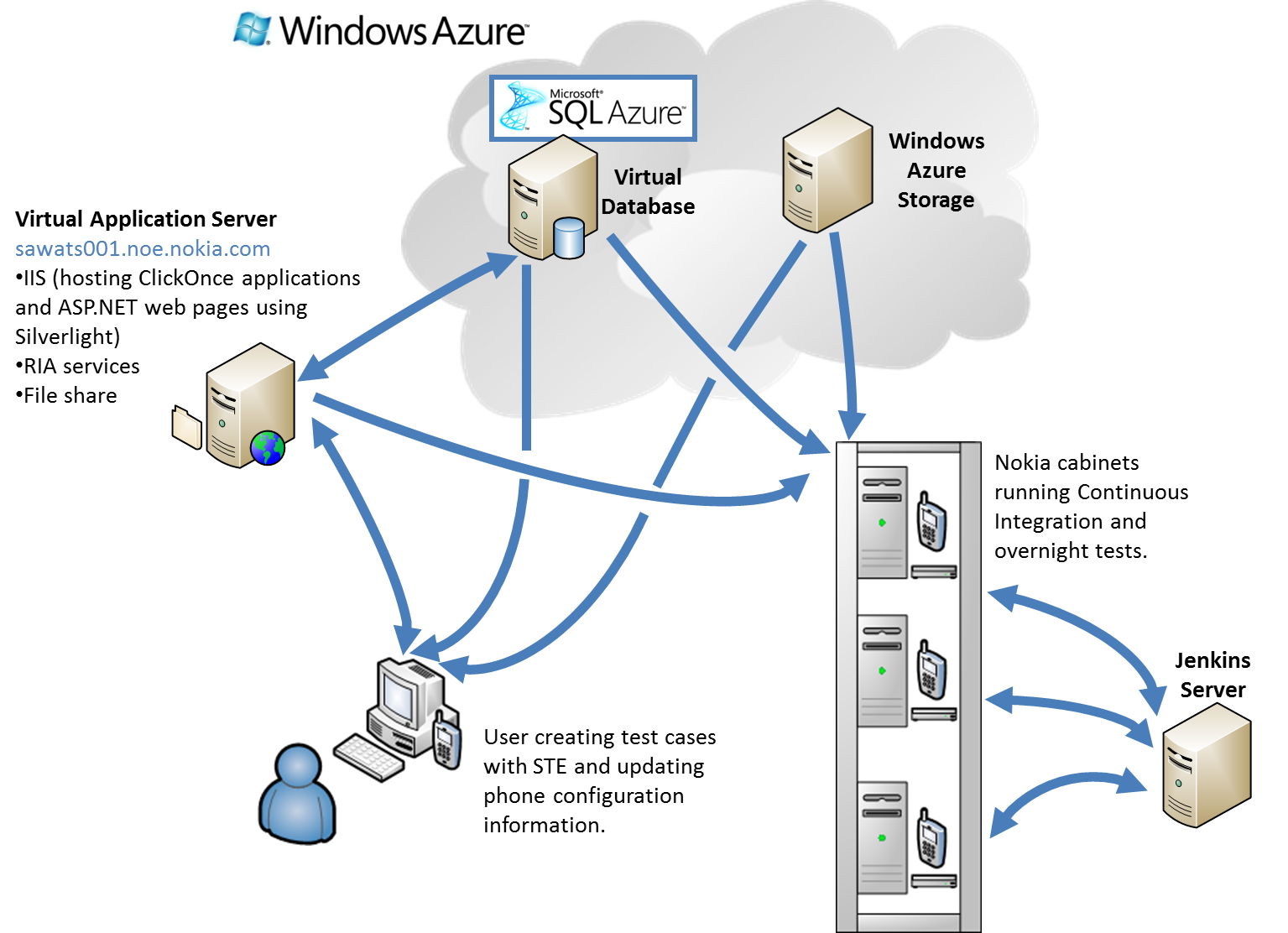


Figure 1 Overview of existing STE/STA architecture

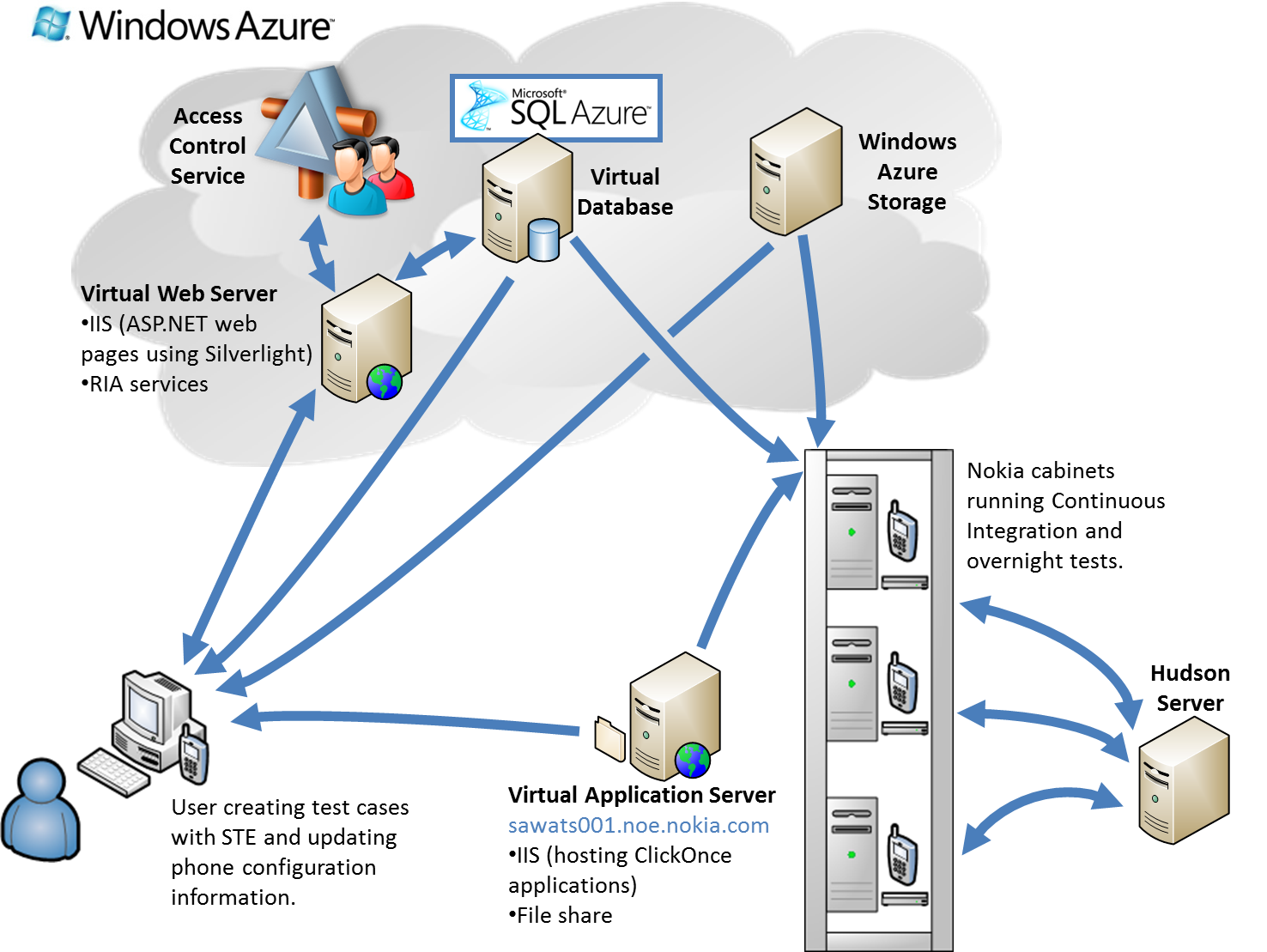


Figure 2 Overview of new architecture

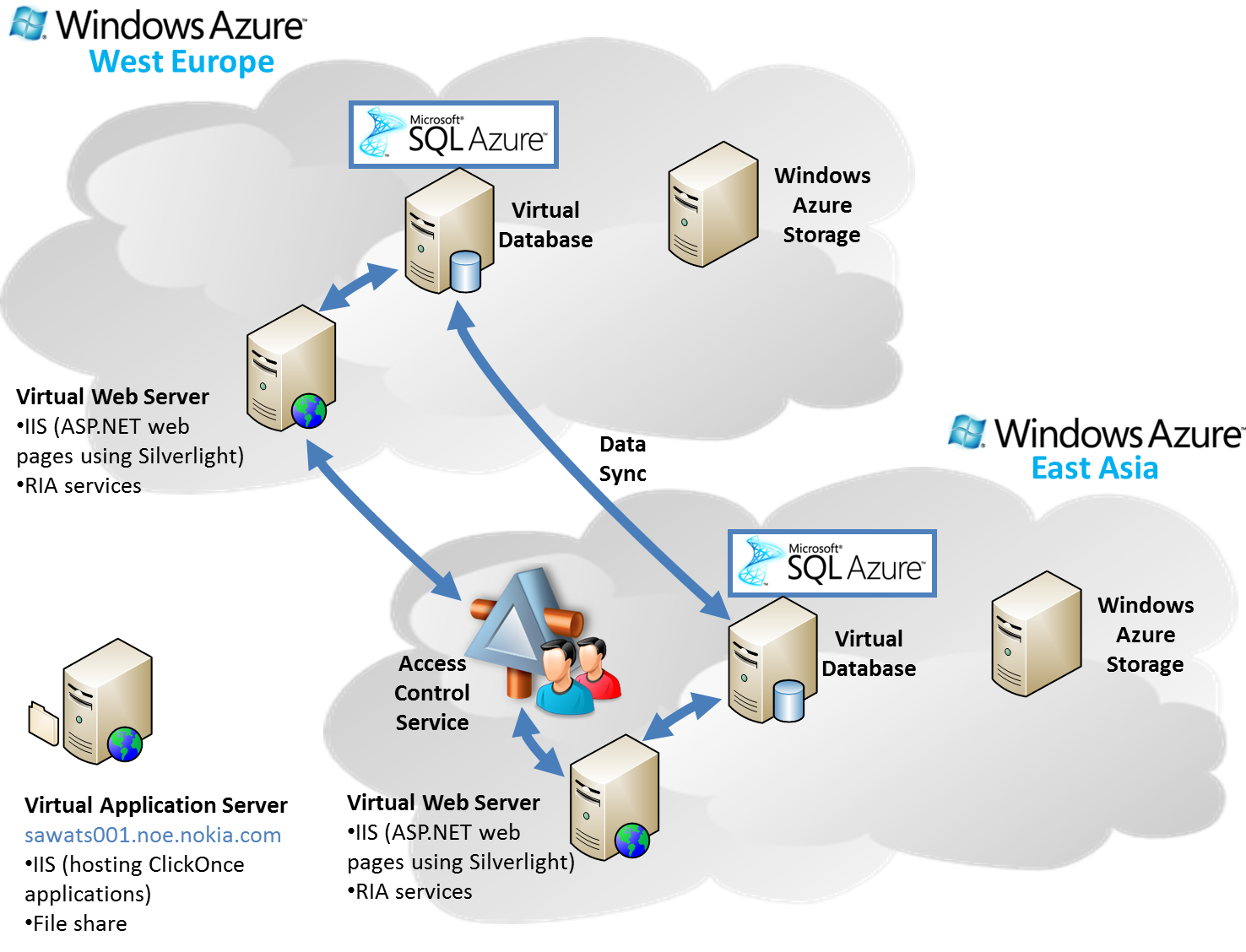


Figure 3 Data Sync between West Europe and East Asia

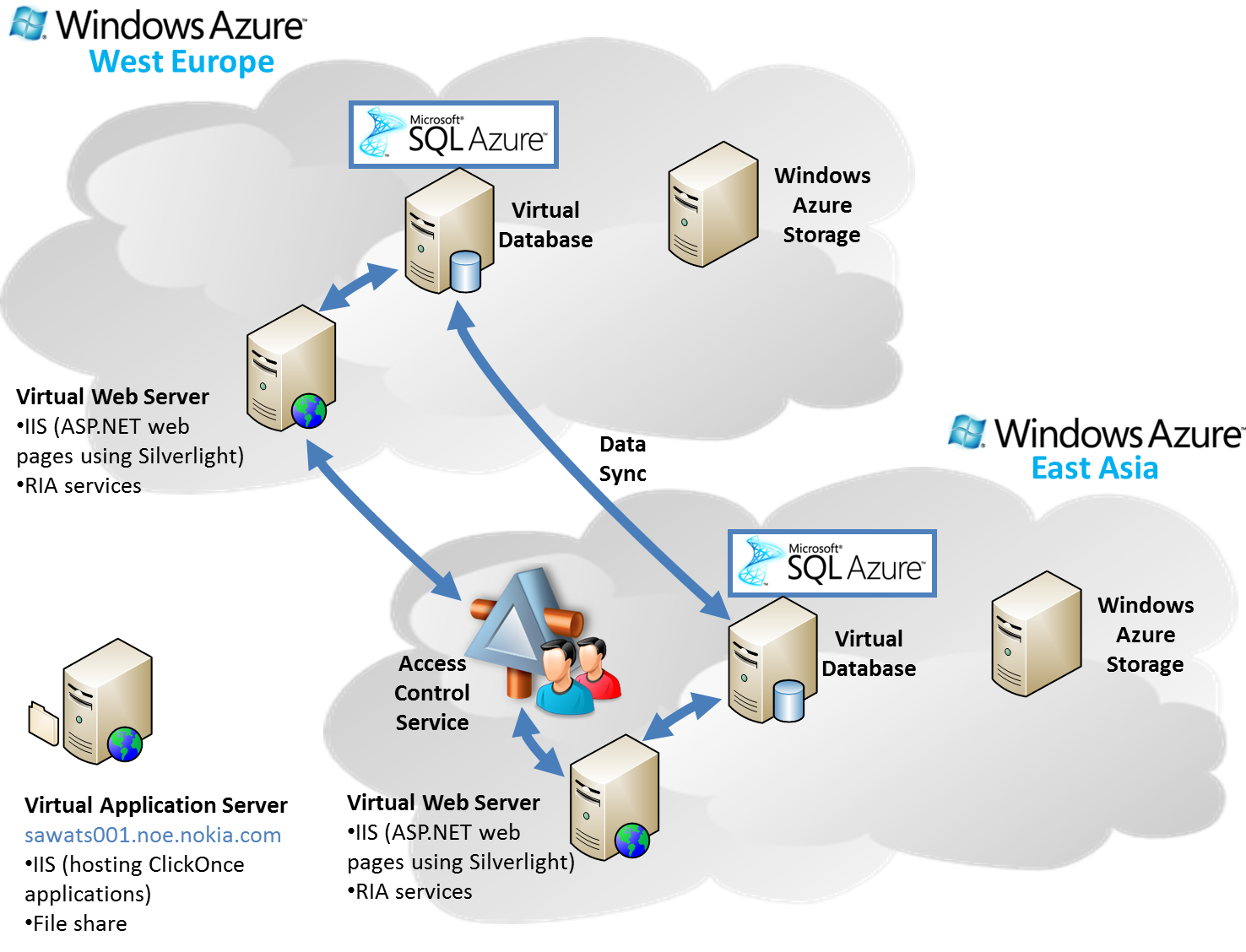


Figure 4 Data Sync between West Europe and East Asia

**Other topics:**

**No overlap** of Granite with STE/STA

Action: Mark to create presentation to explain this. To be reviewed by Lei Li, Wekey and Granite team (someone not on holiday)?   
*Mark could put together a draft presentation and send to Timo P. to review today.*

Plan has always been to port any “good” functionality from STE over to Granite.