# Updates to Interval of Validity Service (IOVSvc)

## ATLAS Software Week
## Dec 1 2003
## Architecture Session

Charles Leggett <Charles.Leggett@cern.ch>

# Updates to Interval of Validity Service

- IOVSvc-00-03-0x and above.
- Conditions Store and Detector Store have been merged.
- Registration of DataHandles and callback functions now handled by StoreGate.
- IOVSvc now has a complete abstract interface (IIOVSvc).

# Data Stores

- As of release 7.3.0, the Detector Store and the Condition Store have been merged.
- Combined store is the **DetectorStore**.
- All conditions and detector data should be accessed via a StoreGate pointer to the DetectorStore

# Registering DataHandles

- The **regHandle()** and **regFcn()** methods of the IOVSvc have been moved to StoreGate. No longer need to get a pointer to IOVSvc to register stuff.
- The syntax has not changed, only the access method:

```
StoreGateSvc *m_pDetSvc;
service( "DetectorStore", m_pDetSvc );


m_pDetSvc->regHandle( handle, key );
m_pDetSvc->regFcn( &Class::callback, obj, handle2,
                   key2 );
```

- Note that if you register a function with a DataHandle, you don't have to register the handle first - it can all be done with one statement.

# Header Files

- The header file that provides the macro for the callback function arguments, "IOVSvcDefs.h" has been moved to the AthenaKernel package.

# Initialization

- For the moment, registering of DataHandles and callback functions should still be done in the initialize() phase if possible.

- For objects such as AlgTools and the like which don't have an initialize() phase, we will soon introduce a "BeginRun" incident. For these objects, registration should occur in BeginRun.

- If you register a callback function after the first event has been processed, you can for the IOVSvc to immediately trigger the callback by supplying an optional extra bool argument (set to true) to regFcn().

# Abstract Interface

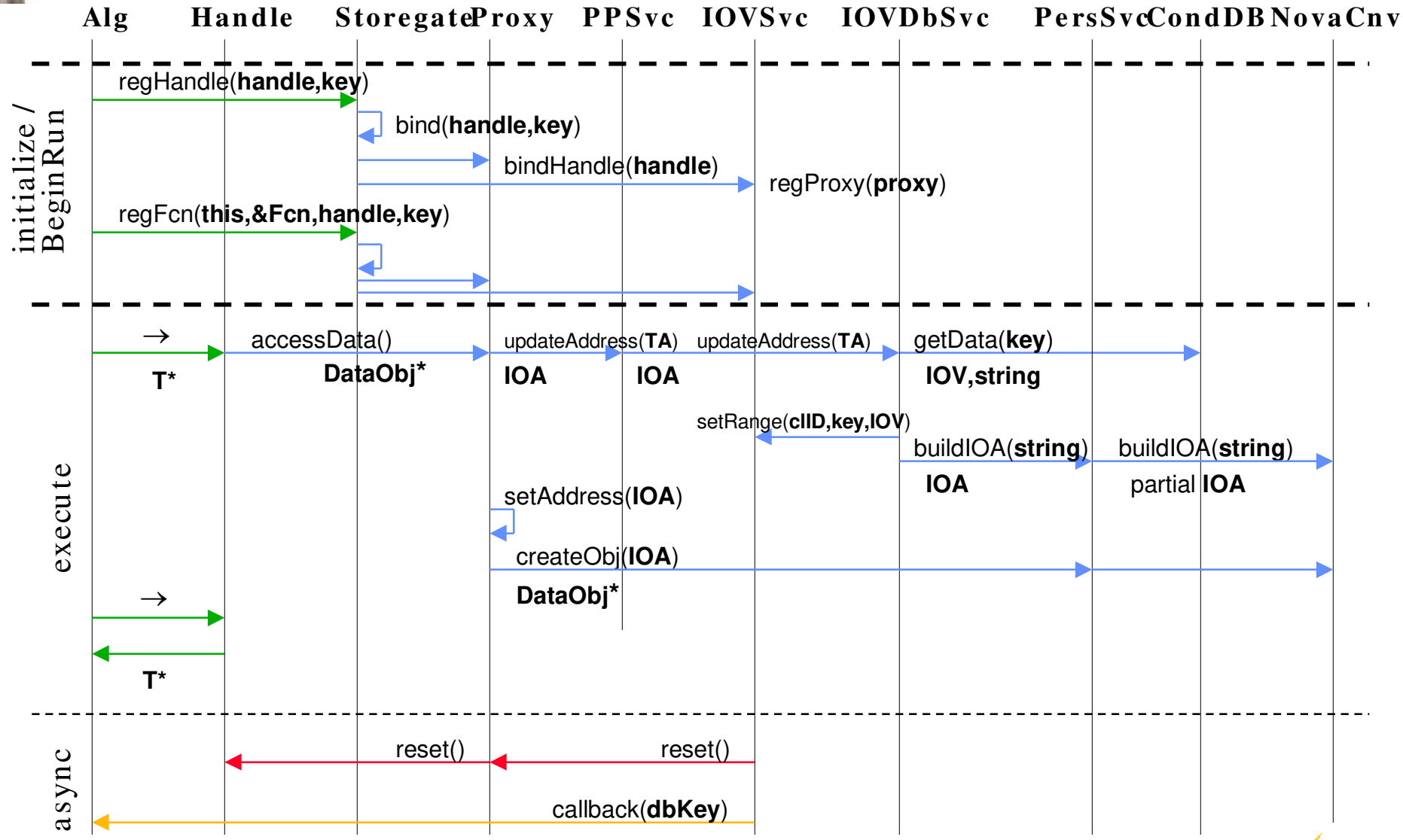
- For all other accesses to the IOVSvc, get hold of service via it's abstract interface:

```
IIOVSvc *m_pIOVSvc;

service("IOVSvc", p_IOVSvc);

IOVRange range;
m_pIOVSvc->getRange(clid, key, range);
```

# Sequence Diagram



# Other Stuff

- Full online documentation:
  - **http://annwm.lbl.gov/~leggett/Atlas/IOVSvc**