Tango To Tine Gateway

- Outline
  - Why do we need gateways?
  - The mapping schema
  - Implementation
  - Example : Archive

Why do we need gateways

- **Tine To Tango**
  - Petra3 experiments need to access machine data
    - Undulator gap and taper, interlocks, beam current, beam position, ...
- **Tango To Tine**
  - Petra3 experiments want to use the machine archiving system
    - Motor positions, temperatures, ...

The Mapping Schema

- Tine server <-> Tango class
  - All devices have the same interface
- Tine device <-> Tango device
  - Addressable on the network with a three field naming schema (x/y/z)
- Tine properties <-> Tango attributes and commands
  - Mapping is possible for all standard data types

Implementation

- Tine To Tango
  - The full Tine device name is mapped as Tango device
    - context/server_name/device_name
  - Properties are mapped to Tango attributes
    - The same names are used
  - Properties with different input and output data types are mapped as Tango commands
    - The same names are used

Implementation

Implementation

- Tango To Tine
  - Tango class name is used as Tine server name
  - A list of Tango devices (or all) of the same class can be mapped into one gateway server
  - The Tine device name field shows the full Tango device name
    - domain/family/member

Implementation

- Tango To Tine
  - Tango commands and attributes are mapped to Tine properties
    - The same names are used
  - Tine context and subsystem can be chosen for the gateway

Implementation

Example : Archive