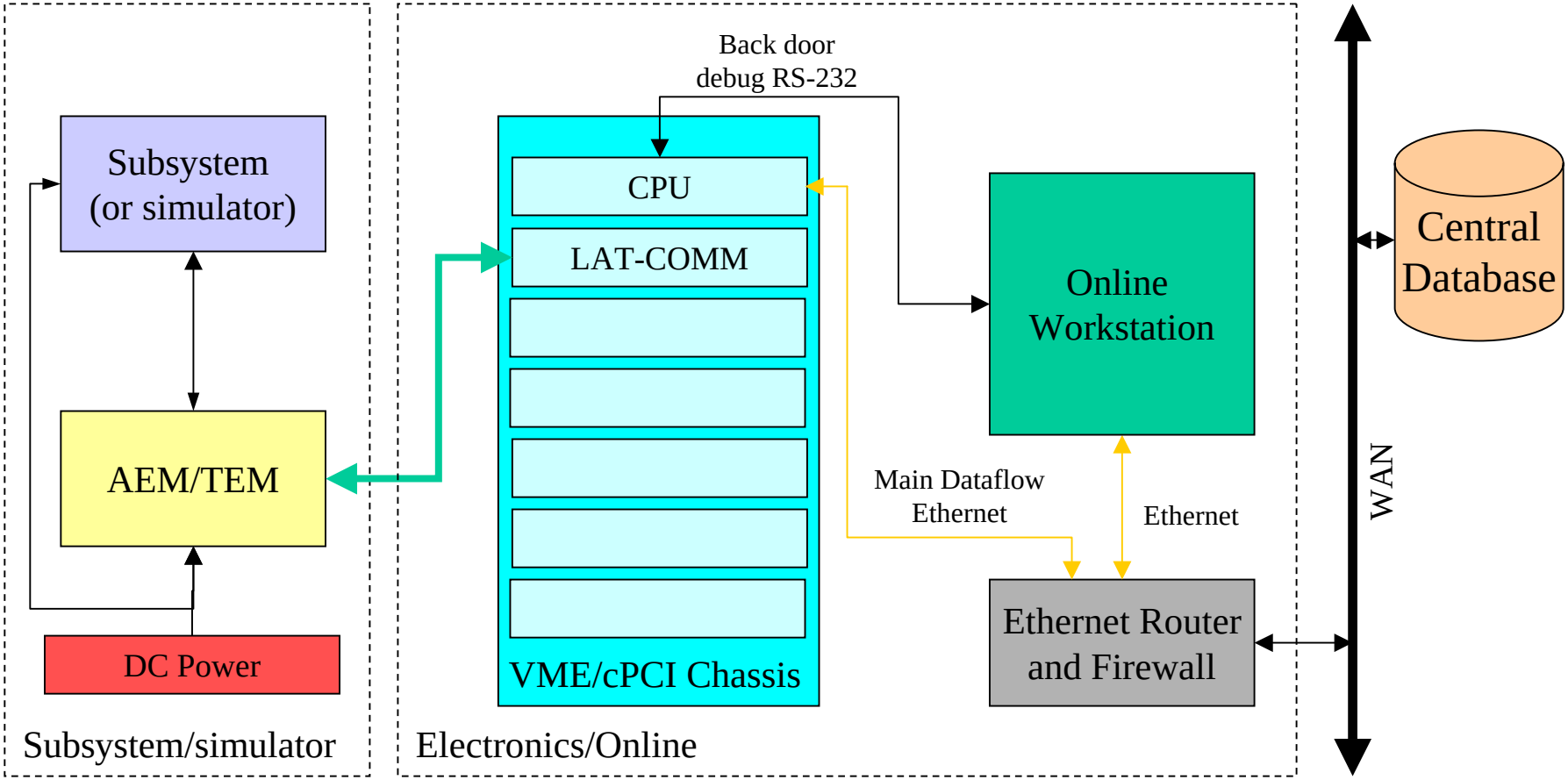
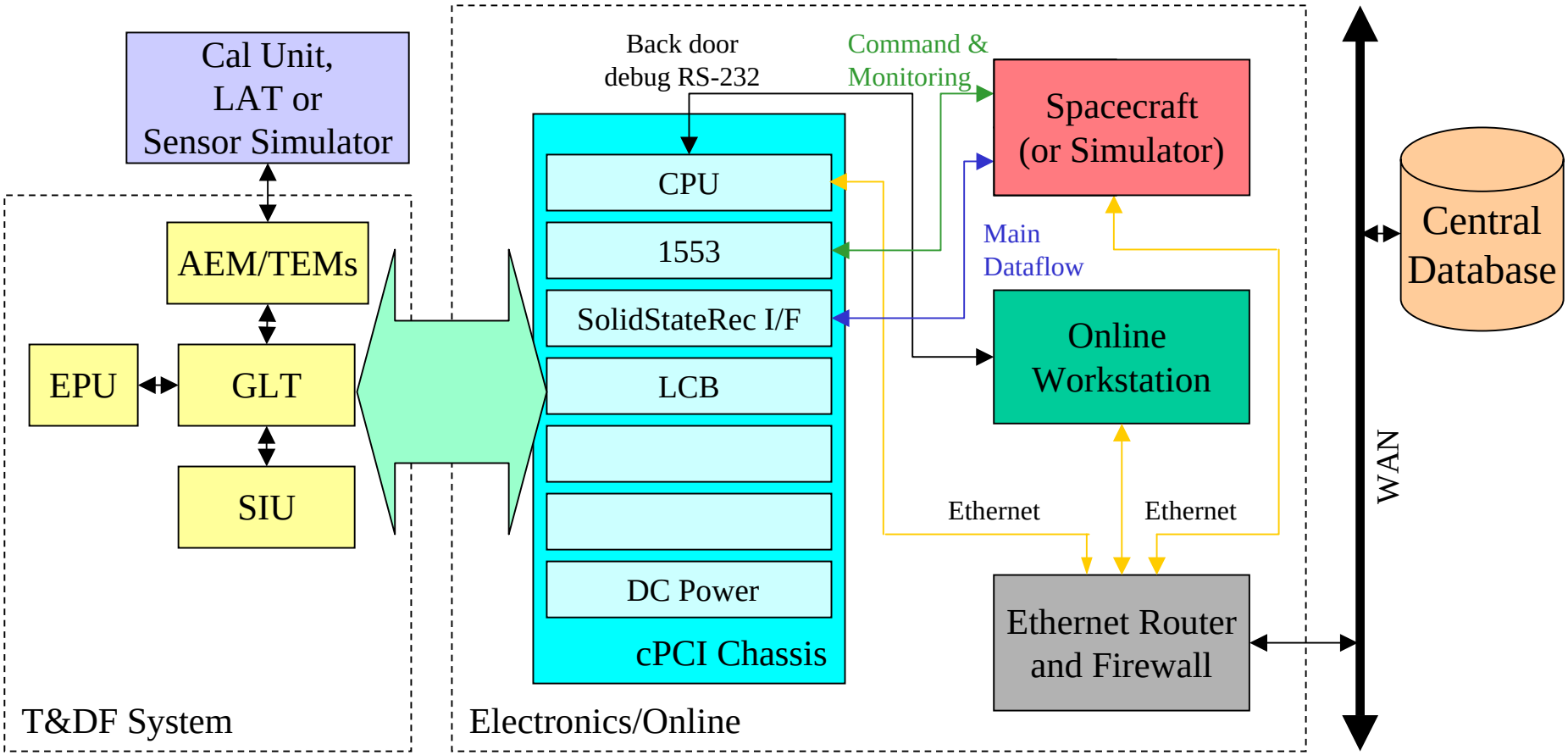
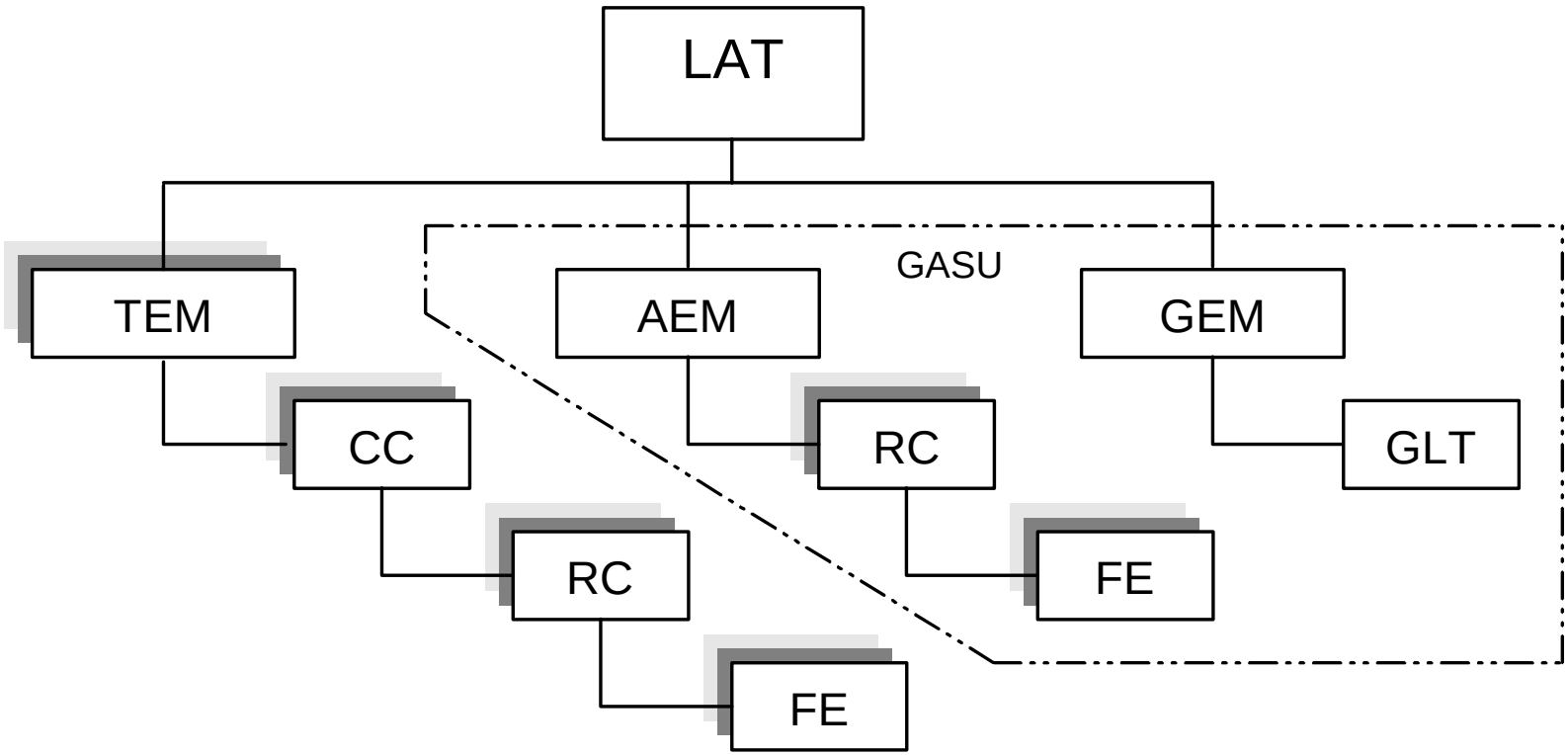
EM1 EGSE Configuration



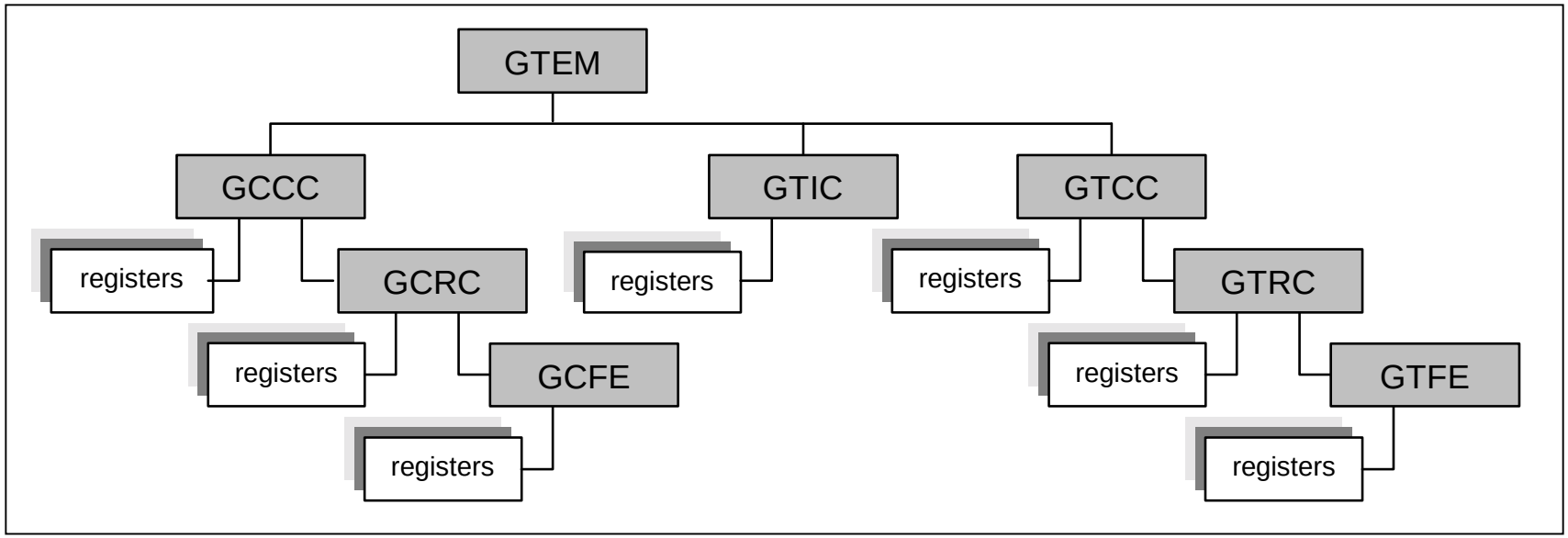
EM2, CU, FU EGSE Configurations



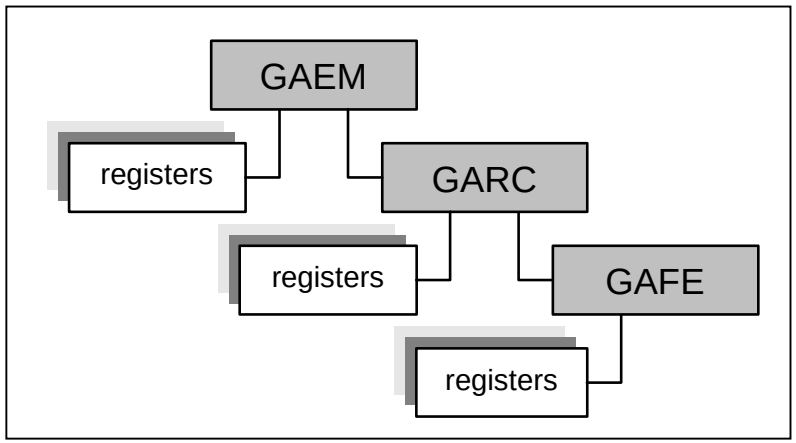
Hardware hierarchy



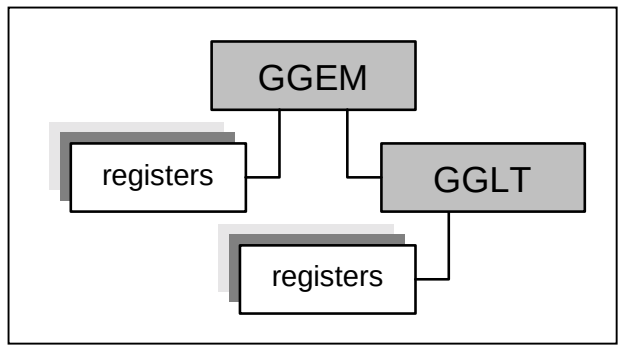
Software hierarchy



Tower Electronics Module



AntiCoincidence Detector Electronics Module



Global Trigger Electronics Module

Quantities (1)

TEM Functional Block	Total Number per TEM	Number of Registers per Block
GTEM Tower Electronics Module	**1**	**6 x 32-bits**
GCCC Calorimeter Cable Controller	**4**	**7 x 32-bits**
GCRC Calorimeter Readout Controller	**16 = 4 x 4**	**8 x 16-bits**
GCFE Calorimeter Front-End ASIC	**192 = 16 x 12**	**3 x 16-bits**
GTCC Tracker Cable Controller	**8**	**6 x 32-bits**
GTRC Tracker Readout Controller	**72 = 8 x 9**	**2 x 64-bits**
GTFE Tracker Front-End ASIC	**1728 = 72 x 24**	**5 x 64-bits**
GTIC Trigger Interface Controller	**1**	**18 x 32-bits**
Totals:	~2000	~10000

Quantities (2)

AEM Functional Block	Total Number per AEM	Number of Registers per Block
GAEM ACD Electronics Module	**1**	**5 x 32-bits**
GARC Calorimeter Readout Controller	**12**	**43 x 16-bits**
GAFE Calorimeter Front-End ASIC	**216 = 18 x 12**	**11 x 16-bits**
Totals:	~200	~3000

1 LAT = 16 TEMs + 1 AEM + 1 GEM

= ~160000 + ~3000 + few registers

= ~165000 registers

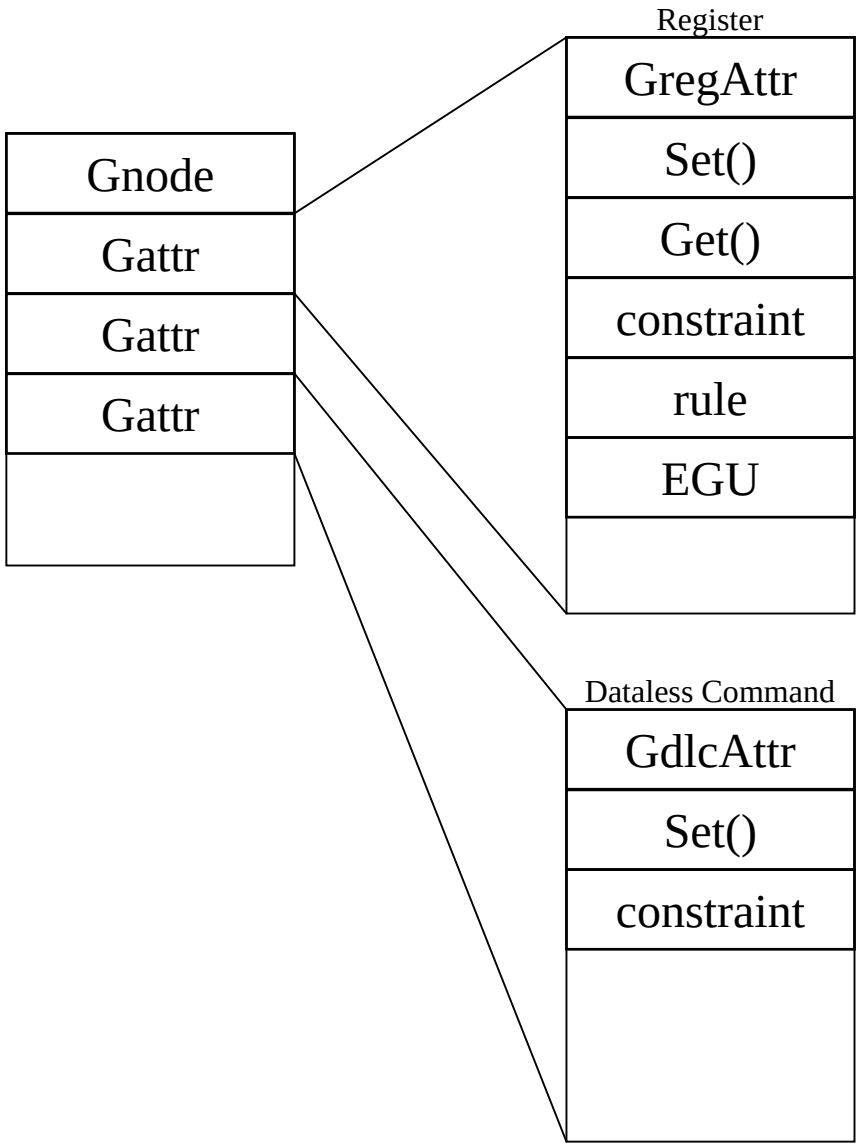
Nodes and attributes

- **Gnode: Base class describing a major block in the LAT hierarchy**
  - **Subclassed to create GLAT, GTEM, GCCC, GCFE, etc. nodes**
- **Gattr: Base class describing attributes of a Gnode**
  - **Examples are registers, dataless commands, etc.**
  - **Some Gattrs can have an associated constraint, rule and or raw/engineering unit conversion class**

Constraints, Rules and EGU

- **Gconstraint: Base class for describing a constraint**
  - Constraints are evaluated when attempting to *write* a mnemonic
  - Two types of constraint subclasses are currently supplied
    - GconstraintLimitRE: RE => Raises Exception when attempt is made to violate limits.
    - GconstraintLimit: Pegs value to limits when attempt is made to violate limits. No indication given to caller that violation attempt was made.
- **Grule: Base class for describing a rule**
  - Rules are evaluated when a mnemonic is *read*
  - One example rule subclass is currently supplied
    - GRuleLimit: When value is outside limits, a message is printed
- **GEGU: Base class for converting between raw and engineering units**
  - One example subclass is currently supplied
    - GEGU_linear: Provides linear conversions
  - Could be used to convert between values and state names

Nodes and Attributes

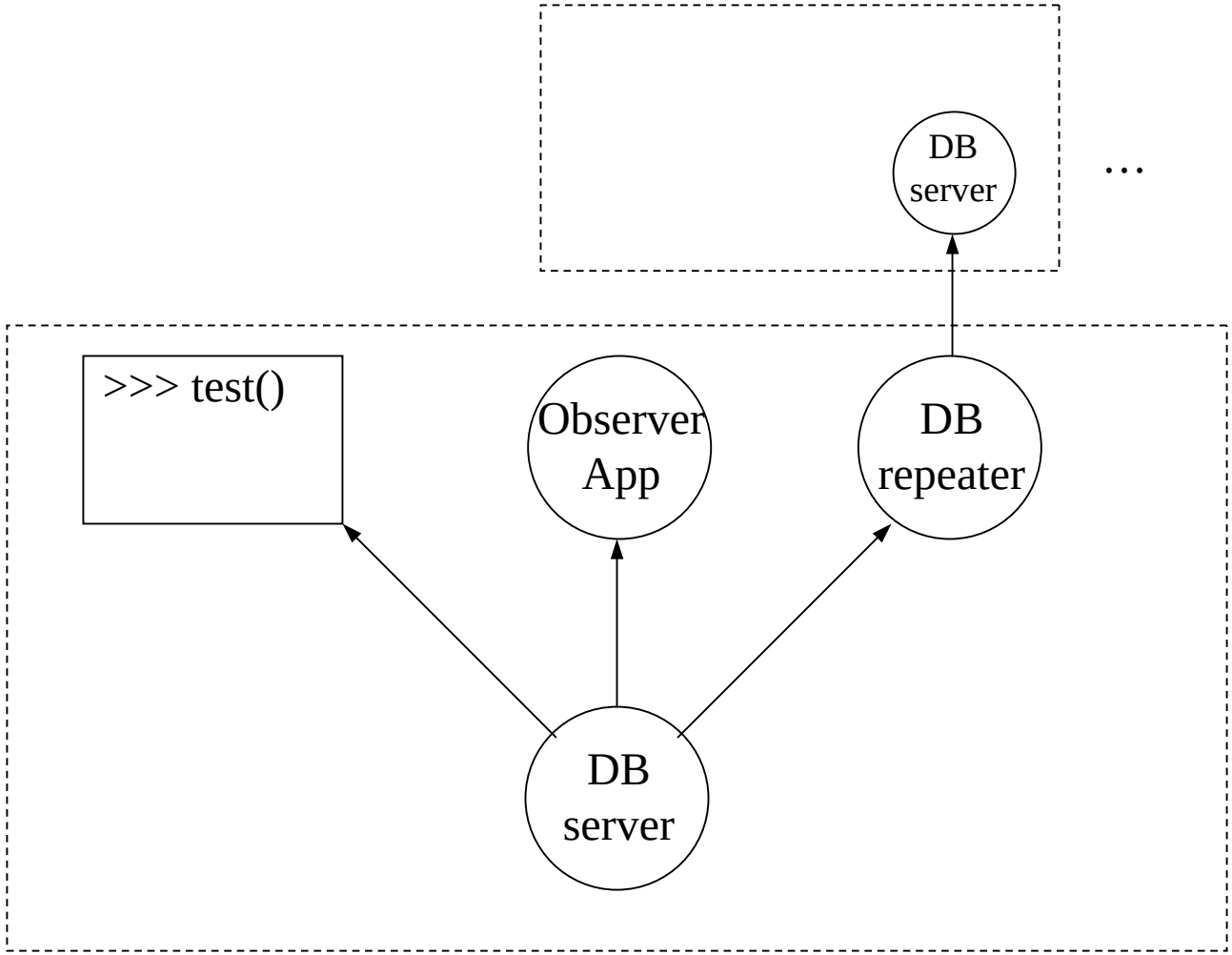


- **Set() evaluates constraint in *engineering* units**
- **Set() then converts to raw units before loading the value on the hardware**
- **Get() reads the hardware and converts the raw value to engineering units**
- **Get() then evaluates rule in *engineering* units**
- **If no EGU is defined, raw and engineering units are the same**

Cautions

- **Beware of naming inconsistencies**
  - GlimitConstraint should be GconstraintLimit, etc.
  - We'll try to get these fixed as we go along
- **Beware of constraints and rules**
  - These are evaluated directly in mnemonic accessors
  - Don't provide one if it is not needed
  - Ensure that evaluation is success oriented and fast
    - Violations should not happen frequently and are thus not as CPU time critical
  - Minimize memory usage
  - Attempt to be generic so it can be reused with multiple attribute instances
- **Beware of rounding issues with GEGUs**
  - Constraints and rules are evaluated using engineering units

Software Architecture



Migration

- **EM-1: No FSW beyond hardware drivers**
  - Will be used for the EM test (cosmics, Van de Graaff photons)
- **EM-2: Development platform for multi-tower support**
  - Embedded systems run FSW code
  - Commanding will be done through a more realistic dictionary
  - Event format changes from TEM/AEM output style
  - Test bench scripts will still be able to be run
- **CU: Four tower system used for the Beam Test at SLAC**
  - No ACD contribution
  - Evolution of EM-2
  - Will need to handle external sources of data
  - 1553, SSR and SIS communications not required to satisfy test
  - Test bench scripts will still be able to be run
- **FU/LAT: The complete system**
  - Communications *only* through SIS, 1553 and SSR
  - Will need IOC/MOC-like interface
  - Test bench scripts will *not* be able to be run

Schedule

- EM test
  - March/April 2003
- CDR
  - April 2003
- SIS (Spacecraft Interface Simulator)
  - April 2003 (preliminary version January ’03?)
- CU beam test
  - May/June 2004
- FU/LAT integration
  - October 2004
- Airplane end-to-end test
  - February 2005