# IM_VTC_20100908

- see Google presentation

    - models of database design
        - loosely coupled individual with links by logic in applications
            - e.g. - can't add to siteDB until are in Personnel database - enforced by software
            - flexible - can alter modules as long as web service IO stays the same
            - easy to extend
            - easier to split up development work
            - requires web services recommendations
                - they need to play nicely together
        - integrated - tightly coupled
            - allows foreign keys and joins
            - enforces consistency without business logic
        - neither one precludes using CMS as client or use with PASTA
        - discussion
            - need to be realistic - this is not a "white room" redesign
                - model to beat is loosely coupled - works better on moving forward existing systems
                - scope differences
                    - if biblio and personnel are tightly coupled - need personnel entries for everyone associated with publications
                    work with other groups will lead to some scope issues
                        - eg if work with NEON do we need to ingest all NEON poeop
            - these are not exclusive options
                - could have siteDB and personnel integrated, but loosely coupled to biblio
                - if went integrated may want to simply hire company that specializes in enterprise databases
            - advantage of loosely coupled - can move modules forward one at a time
            - important to avoid redundant and out of sync infomation - need to decide what goes where
                - share keys across databases, not actual content
            - in loosely coupled need to figure out how we will implement pseudo-foriegn key
            - one business rule we may need is that nothing ever gets deleted - just marked as inactive
    - IM meeting
        - working groups - 3 to choose from

- Wed afternoon
  - everyone looking at Net database
  - yesterday
    - could split into gathering requirments and use cases
      - end user interfaces
      - service interfaces for developers
      - models for database desigh
        - map out relationships between data types
  - would prefer a common charge for 3 or 4 groups - get vision - what should system look like
    - provides use cases
    - later sessions can follow up on
    - do NOT want to do around the room!
    - would focus on requirements at ALL levels
    - follow ons would address database questions
  - questionaire with 3 to 5 questions to lead the discussion
    - general - what are goals
      - then perhaps explore more specifics
      - what are the challenges
    - want to get EVERYONE involved
  - look at use cases - what can we DO with the databases
    - want to anticipate what scientists will want
    - discuss with PIs how they use net databases
  - extent to which sites will use network capabilities vs site duplication
    - where some of these efforts have already been made at site
      - network DB can't provide LESS service than site service
  - how do we integrate back into the LNO databases - synching site databases
    - example - units database - bought some sites out for training 1:1
    - also need some training on web service clients
  - web services - plan is for read and write capable
    - could have synchronization service
    - still would be good to look at entire paraigm - synchronization
  - now use Endnote for bibliographic database
    - would depend on what sites want to continue or get away from it...
    - parts of requirements gathering
    - usually one of 2 or 3 solutions will meet site needs
      - some sites operate in larger context - use subsets to populate LTER

- this is a good example - have examples of harvests etc. using Endnote as exchange service
- are there existing exchange formats we can use?
  - examples from unit dictionary, projDB
  - need to look at exchange formats
  - XML provides lots of flexibility
    - can use EML modules as starting point
    - other versions are just a stylesheet away
  - if use RESTful structure - can support multiple exchange formats
    - unit dictionary does this
- ClimDB, HydroDB another example
  - want to be inclusive of all the databases we have now and plan in the future
  - these will become part of PASTA

- two sessions on Thursday
  - NIS design
  - web services
    - Mason working on outline of questions