- **Physics Driver** (Steve Goldhaber, Andrew Gettelman & Francis Vitt)
    - Since the CPD has already been funded, just refer to CPD Proposal
    - Physics group should provide priorities (end of February)
- **Dynamical Core** (John Dennis & Richard Loft)
    - Ability to run single precision without restricting parameterizations (this is also listed under future architectures)
    - Dynamics-physics coupling: Cross-group session will be convened to discuss requirements
    - Granularity of precision types in dycores
- **Data Assimilation** (John Dennis/Brian Dobbins & Steve Goldhaber)
    - Community models must be DA friendly. One infrastructure issue there is allowing the DA system to define length of next forward-model run.
    - Stable workflow interface
    - Easy to control computation and output of diagnostic quantities
    - Minimize overhead to start / restart / resume forward model
    - Exact (BFB) restart for testing
    - Efficient/scalable ensemble forecasts on available high performance computing including large numbers of tracers.
    - Configurable to be efficient for range of space/time/depth scales including limited-area.
    - Can perform a sequence of short integrations with minimal computational overhead relative to a single long integration.
    - Scripting of multiple jobs as well as single executable ensembles. Use case is ensemble with members running with different physics or with different resolutions.
    - Easy-to-use interface to compute/output diagnostic quantities.
- **Intra-component coupling** (Rocky Dunlap, Mariana Vertenstein & Francis Vitt)
    - (dynamics, physics, chemistry, ionosphere)
    - Regridding, concurrency
    - Requirement? Use case(s)?

- **I/O subsystem** (Jim Edwards & Kevin Paul)
    - end-to-end parallel I/O scalability & performance
    - data compression capabilities of the underlying libraries.
    - asynchronous vs synchronous
    - formatting standards
    - Compatibility with postprocessing workflow

- **Supporting Workflow/Analysis Tools** (Kevin Paul, Sheri Mickelson, Matt Long & Rich Neale)
    - Workflow Automation
    - Pre-run tool enhancement / automation (grids, mapping, initial conditions)
    - Performance (monitoring, tuning, 'coding for performance' training and documentation)
    - Interoperability with model I/O subsystem
- **Support for future architectures, modes of computing** (Richard Loft & John Dennis)
    - Support for mixed precision models
    - Support for heterogeneous architectures
    - Broader tools and compiler support
- **Code verification infrastructure** (Allison Baker, Youngsung Kim, Steve Goldhaber & John Truesdale)
    - Unit tests
        - Community Physics Framework (CPF) will provide unit testing of itself
    - Better functional testing
        - CPF will provide offline functional testing of individual parameterizations or physics suites
    - verification tools e.g. KGEN, PyECT
        - CPF will provide tools to use captured, provided, or analytic data for offline parameterization verification and validation