



# RIB Analysis - Redundant Process Structure and Output Comparison Design

---

**Chase Cotton**

Visiting Scientist

Associate Director for Cybersecurity, Center on Information  
and Communications Systems (CICS)

Electrical and Computer Engineering

University of Delaware

ccotton at udel dot edu



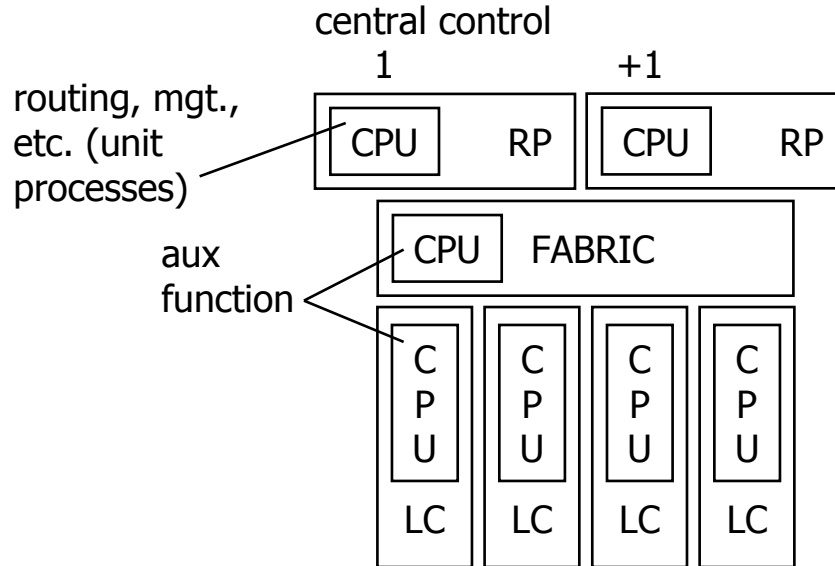
# agenda

---

- redundancy architecture (software)
- output comparison methods
- next steps
- questions

# high level redundancy direction

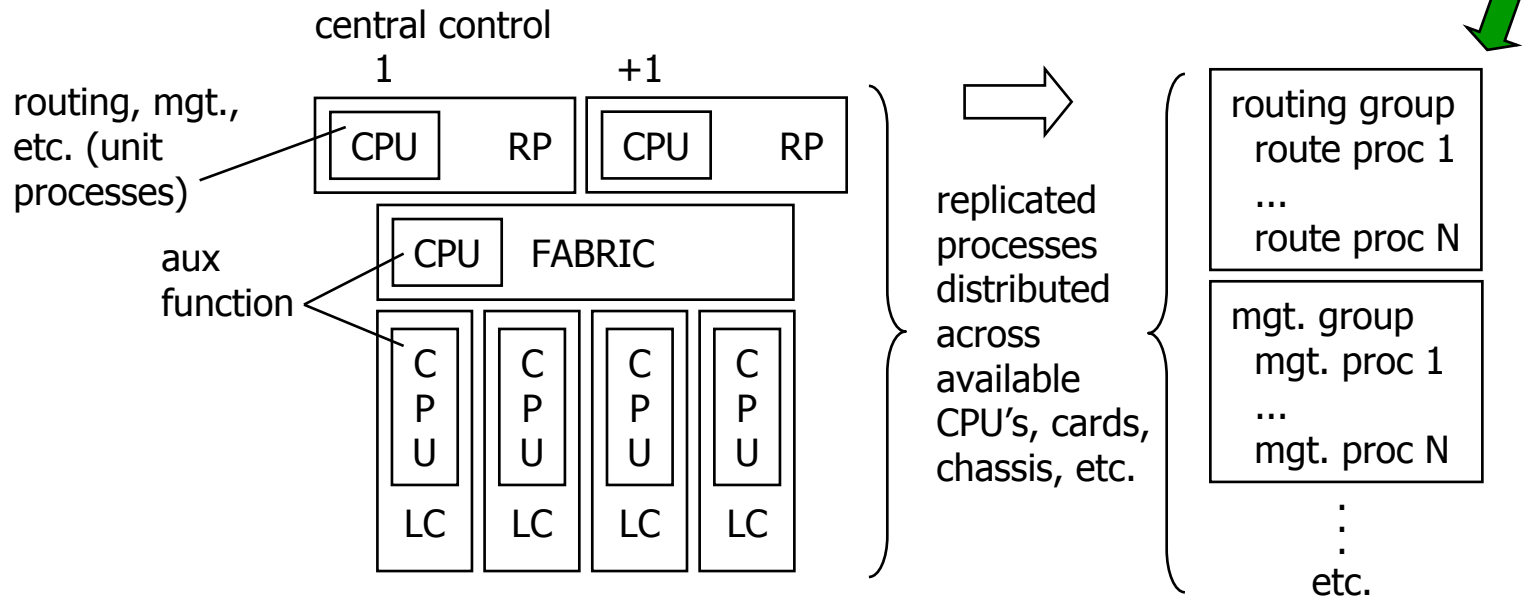
- change model from largely central to redundant



- inadequate:  $1+1$ , hot/cold sparing
  - not inherently bad / wrong, but tendency to take shortcuts leads to a less than robust solution

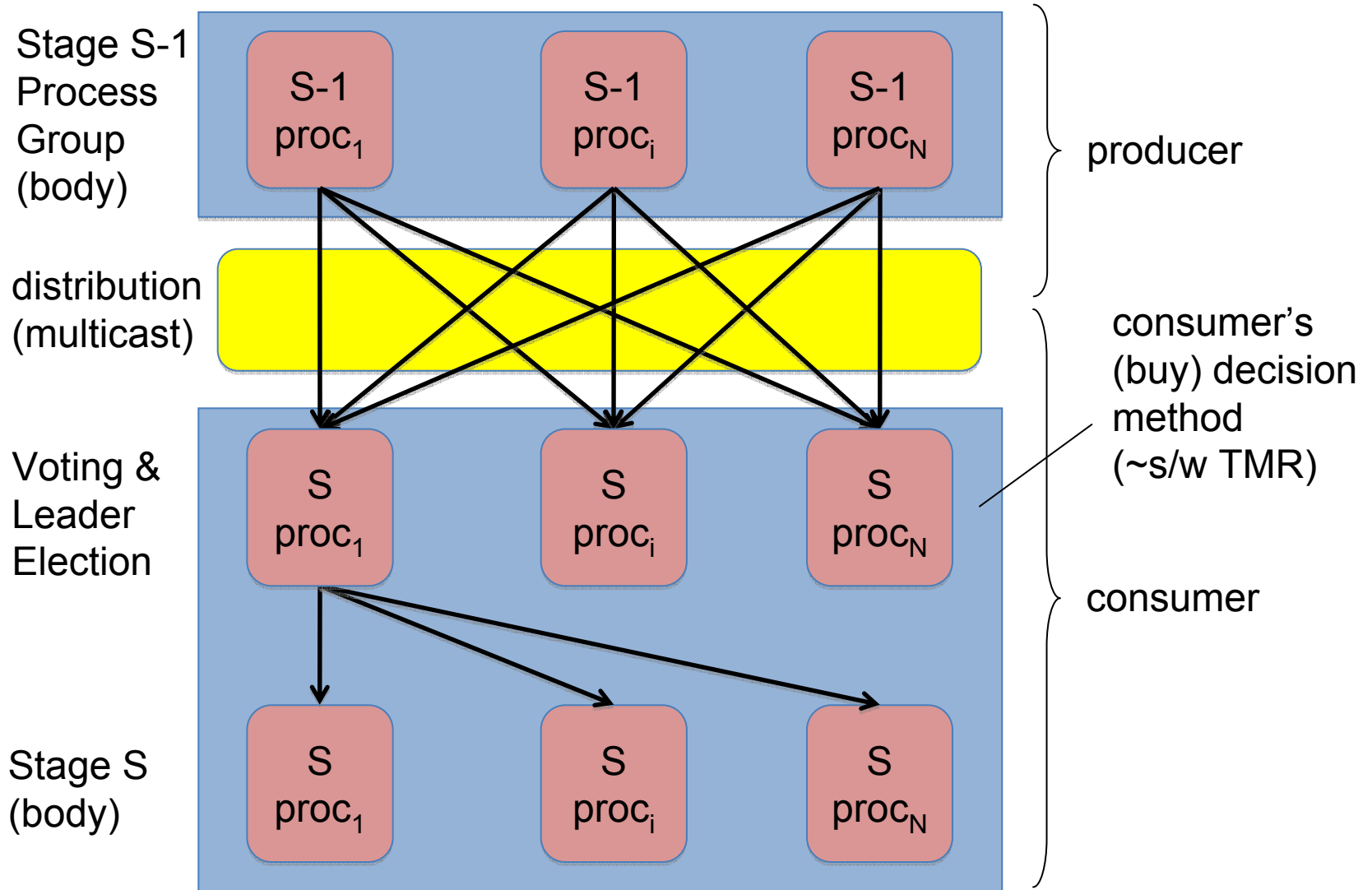
# high level redundancy direction

- change model from largely central to redundant

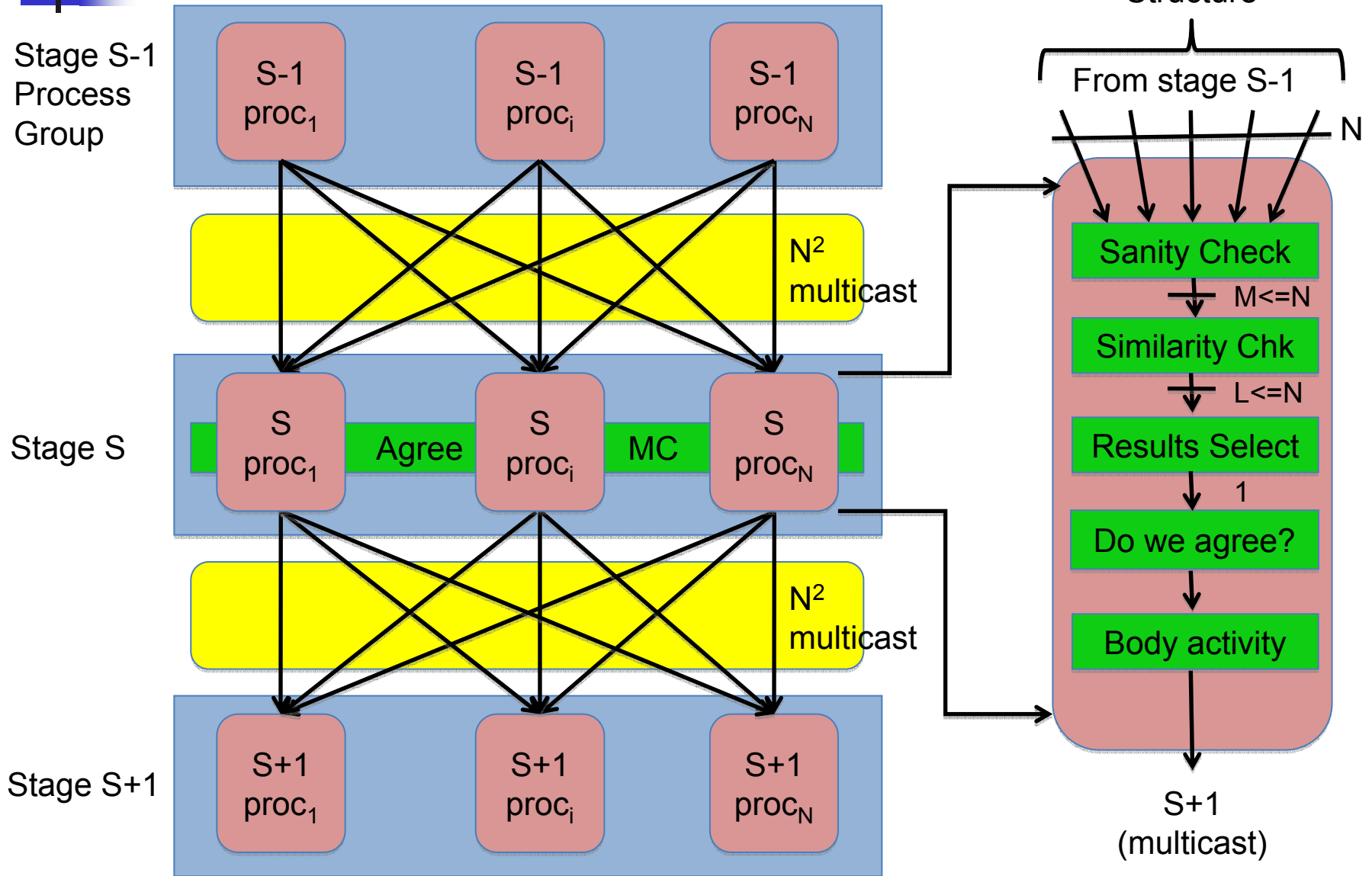


- inadequate: **1+1, hot/cold sparing**
  - not inherently bad / wrong, but tendency to take shortcuts leads to a less than robust solution
- minimum essential: **N-modular redundancy**
  - no exceptions to redundancy – both main functions as well as auxiliary functions (even those supporting the basic redundancy mechanisms, i.e. election, selection, voting)[hard]

# process groups – bazaar model



# process groups – evolution #2



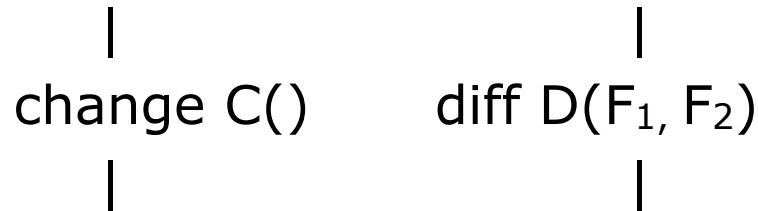


# sanity / self-check / similarity - features

---

## ■ model

body result:  $RIB_1 \rightarrow \text{features}, F_1(RIB_1) \rightarrow \text{n-space (cluster)}$

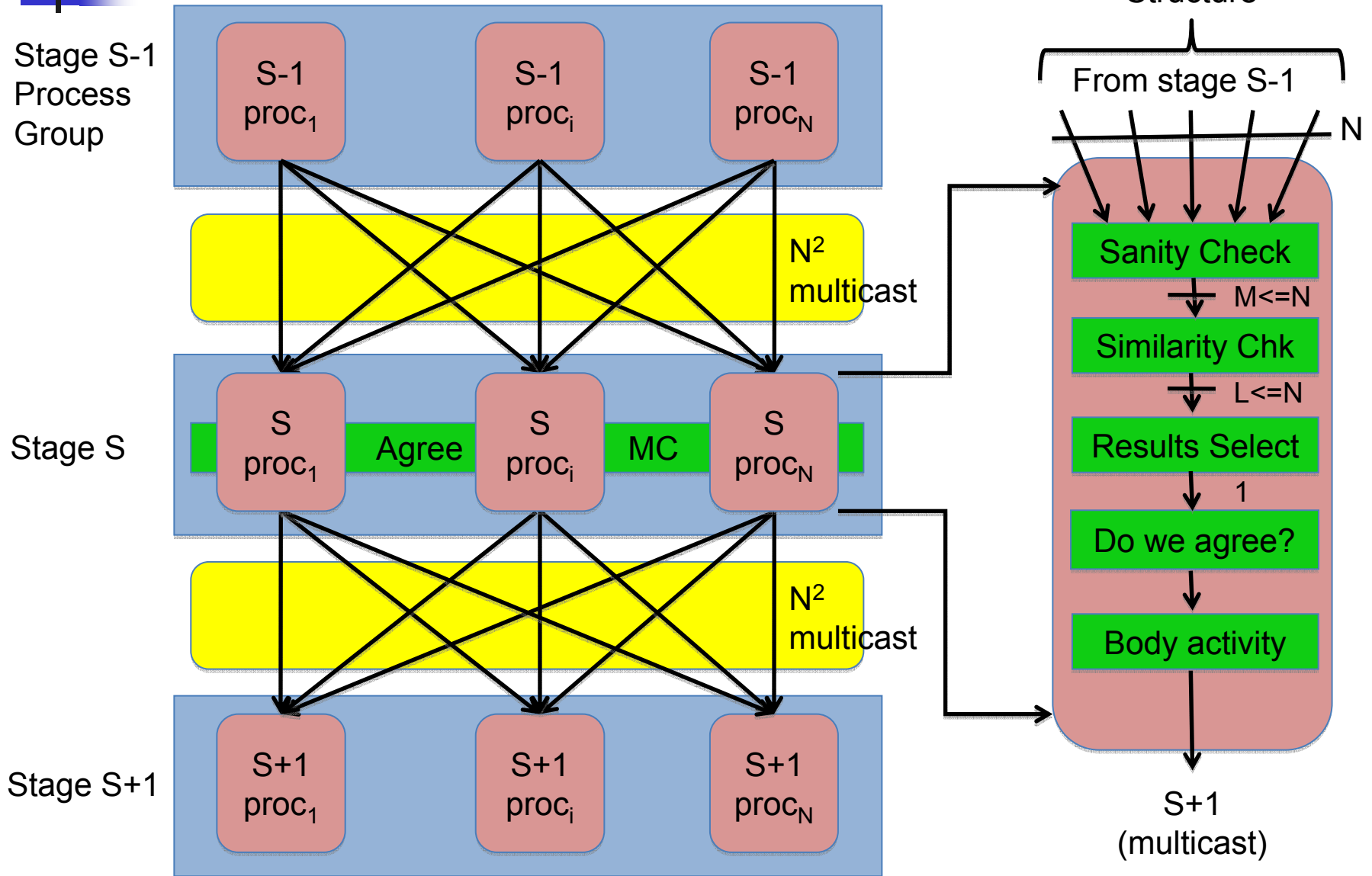


body result:  $RIB_2 \rightarrow \text{features}, F(RIB_2) \rightarrow \text{n-space (cluster)}$

## ■ desire

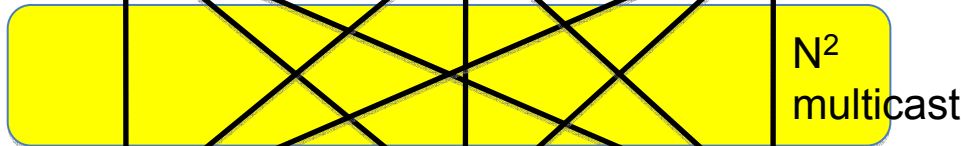
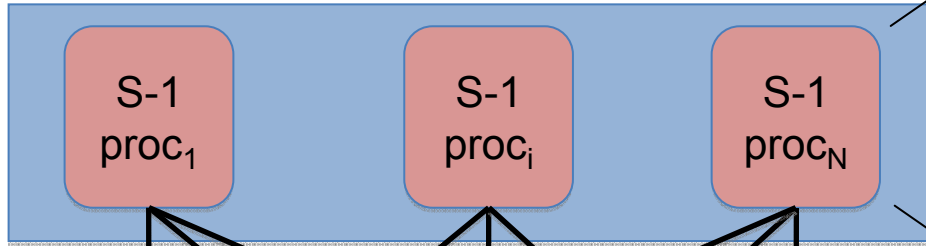
- $D()$  proportion to  $C()$
- $F()$  small in size
- computation of  $F()$  to have minimum cost / complexity
- incremental computation of  $F(R_2 + \text{delta}) < F(R_2)$
- $F()$  and  $D()$  make use of internal RIB computation and data structures and housekeeping (e.g. hashes)

# process groups – evolution #2

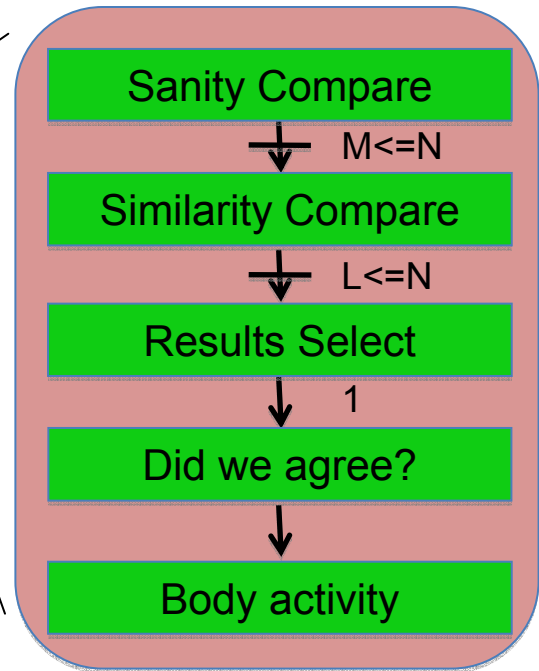
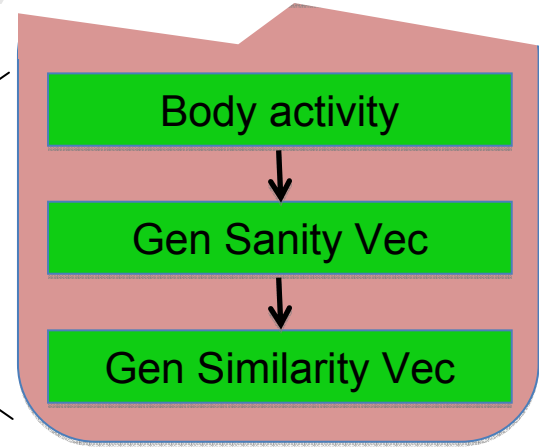
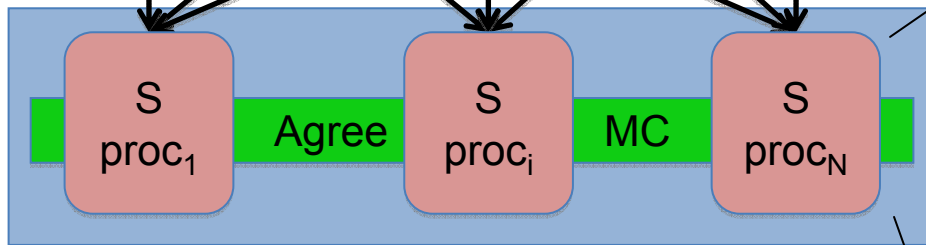


# process groups – current model

Stage S-1  
Process  
Group

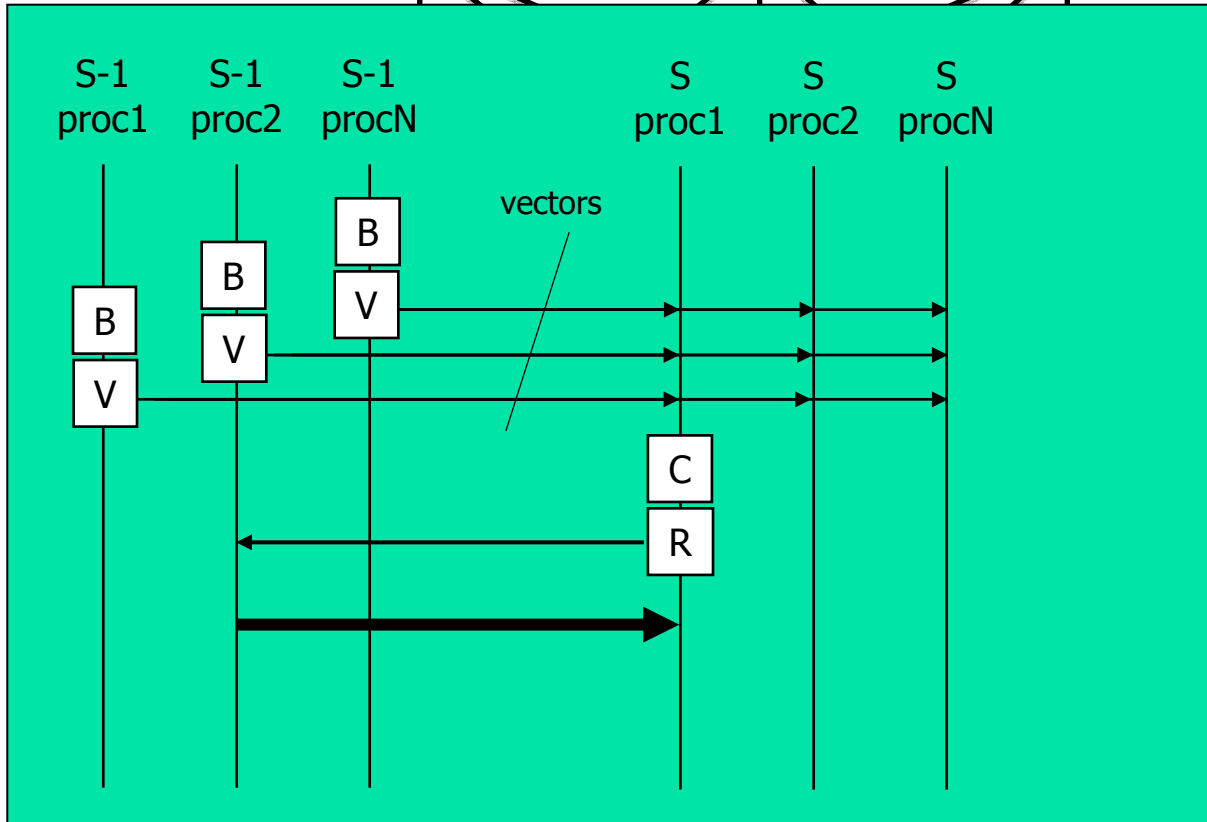
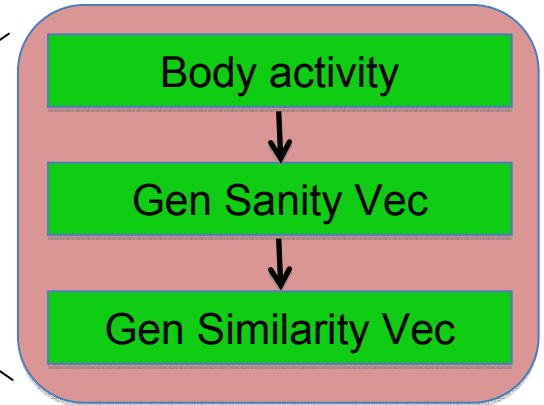
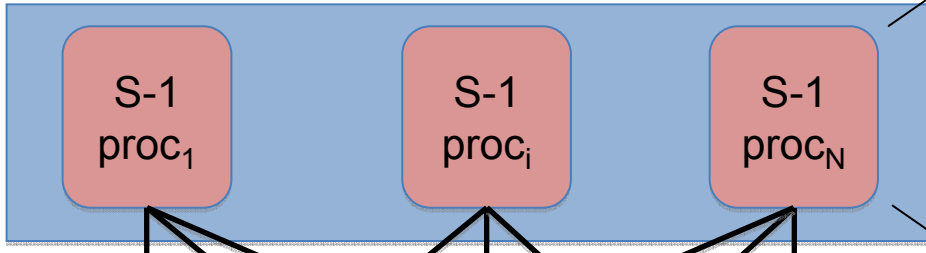


Stage S

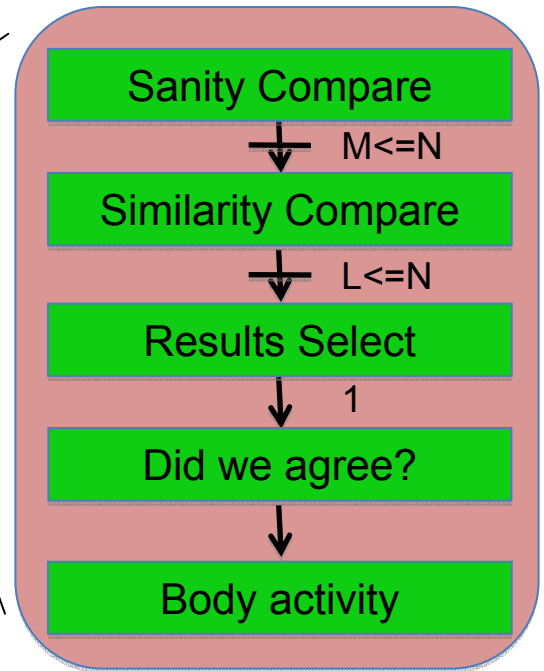


# process groups – current model

Stage S-1  
Process  
Group

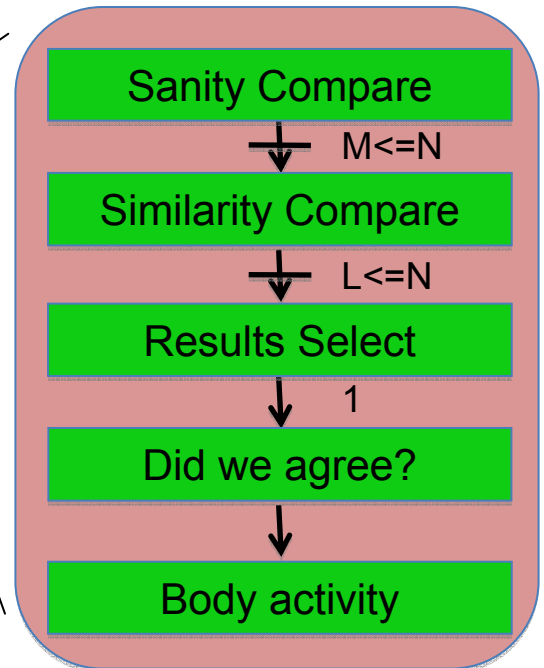
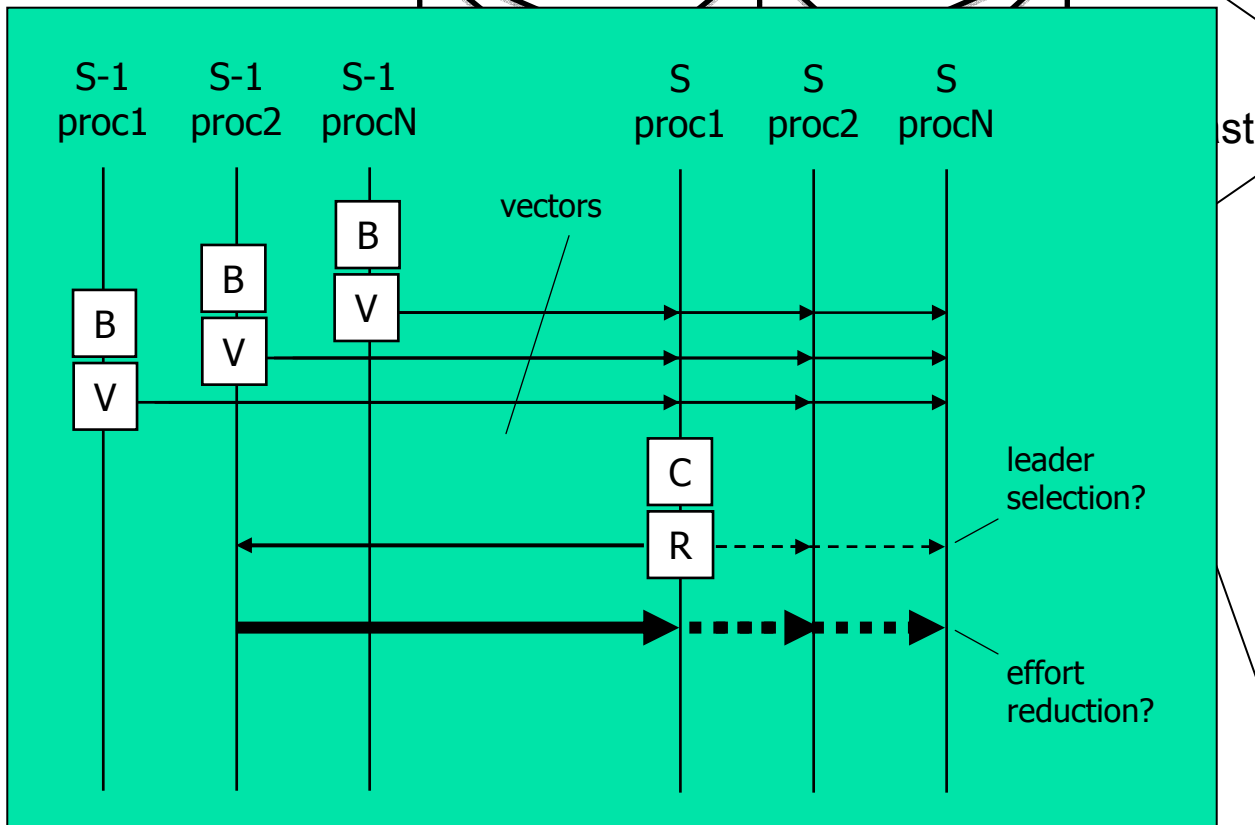
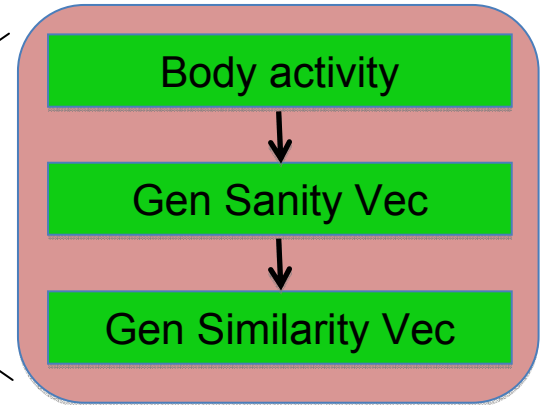
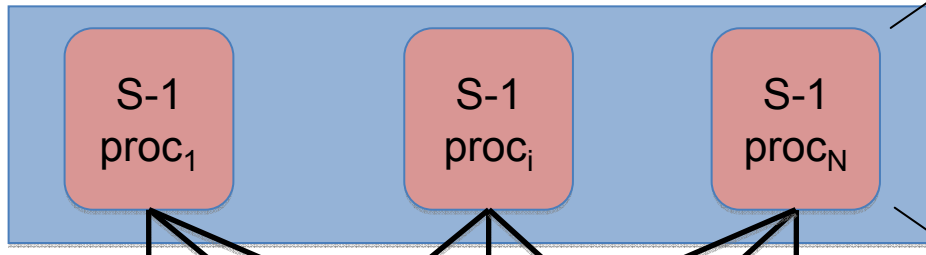


st



# process groups – current model

Stage S-1  
Process  
Group





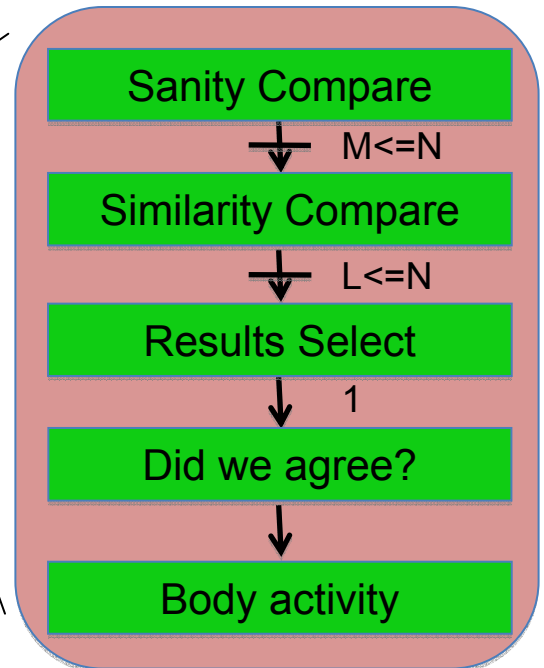
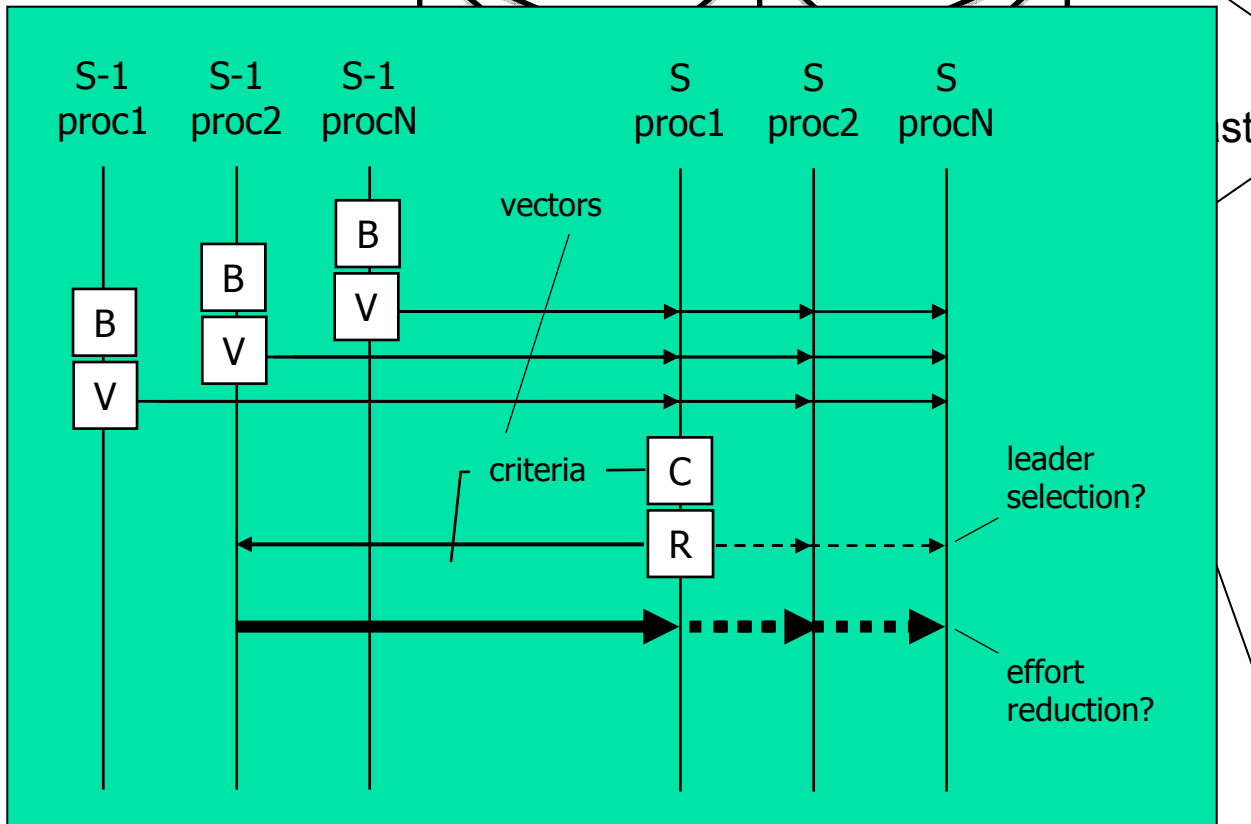
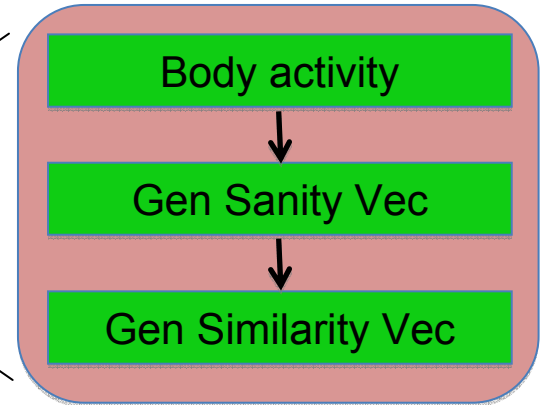
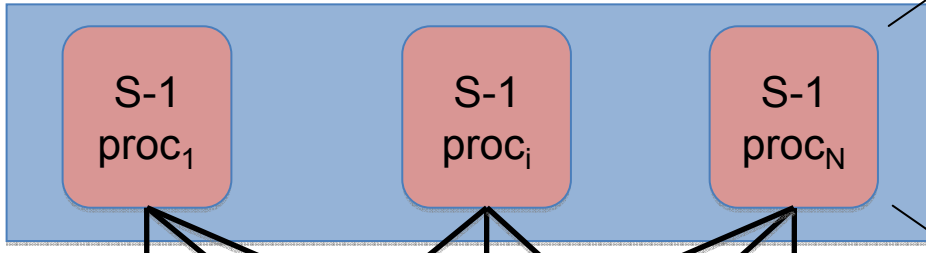
# Vectors and Comparisons

---

- Intent is to drive a process independent redundancy model (hard) vs. re-coding
- Currently exploring routing processes
  - IGP: IS-IS (topology)
  - BGP: eBGP & iBGP (prefixes, AS paths)
  - per protocol and combined RIBs (as a class of outputs)
- Vector methods (BGP)
  - originally – output feature generation with clustering
  - currently segmented for speed/efficiency
    - Sanity / self-check vectors ... “what is clearly wrong”  
 $S = \langle s_1, s_2, s_3, \dots, s_M \rangle$
    - Feature vectors ... “what same, what different”  
 $F = \langle f_1, f_2, f_3, \dots, f_M \rangle$
    - Exact Comparison  
E (scalar) ... “exact same”
  - for clustering  
 $R = \langle S, F \rangle$   
cluster identification (k-means, etc.), distances (differences)

# process groups – current model

Stage S-1  
Process  
Group





## next steps ...

---

- refinement for AS path vectors and associated indicators
    - metrics for sanity – thresholds?
    - cluster distance difference experiments for controlled defects
    - understand viability of matching criteria idea
  - hardening of the redundancy model(s) (protocol/structure)
  - vector software structure
    - test vector (sanity / feature / compare) mechanisms resident with body code
  - and prototyping
  - understand fast path (decisions in  $\leq 200$  mS) and fall back after “committee” (group) decision (if made in error)
- 
- thank you for your time
  - Questions?