NC STATE UNIVERSITY

ENGINEERING ONLINE

Lecture Notes

Course Number: CSC 513

Instructor: Dr. Singh

Lecture Number: 2

Dynamism

Independence of system configurers and administrators

- Sociopolitical reasons
  - Ownership of resources
  - Changing user preferences or economic considerations
- Technical reasons: difficulty of maintaining configurations by hand
  - Same reasons as for network administration
  - Future-proofing your system

TeraGrid

Ocean Observatories Initiative

Coherence

Think of this as an alternative to consistency

- There may be no state (of the various databases) that can be considered consistent
  - Maintaining consistency of multiple databases is difficult
  - Unexpected real-world events can knock databases out of sync with reality
- What matters is
  - Are organizational relationships preserved?
  - Are processes followed?
  - Are appropriate business rules applied?

Integration

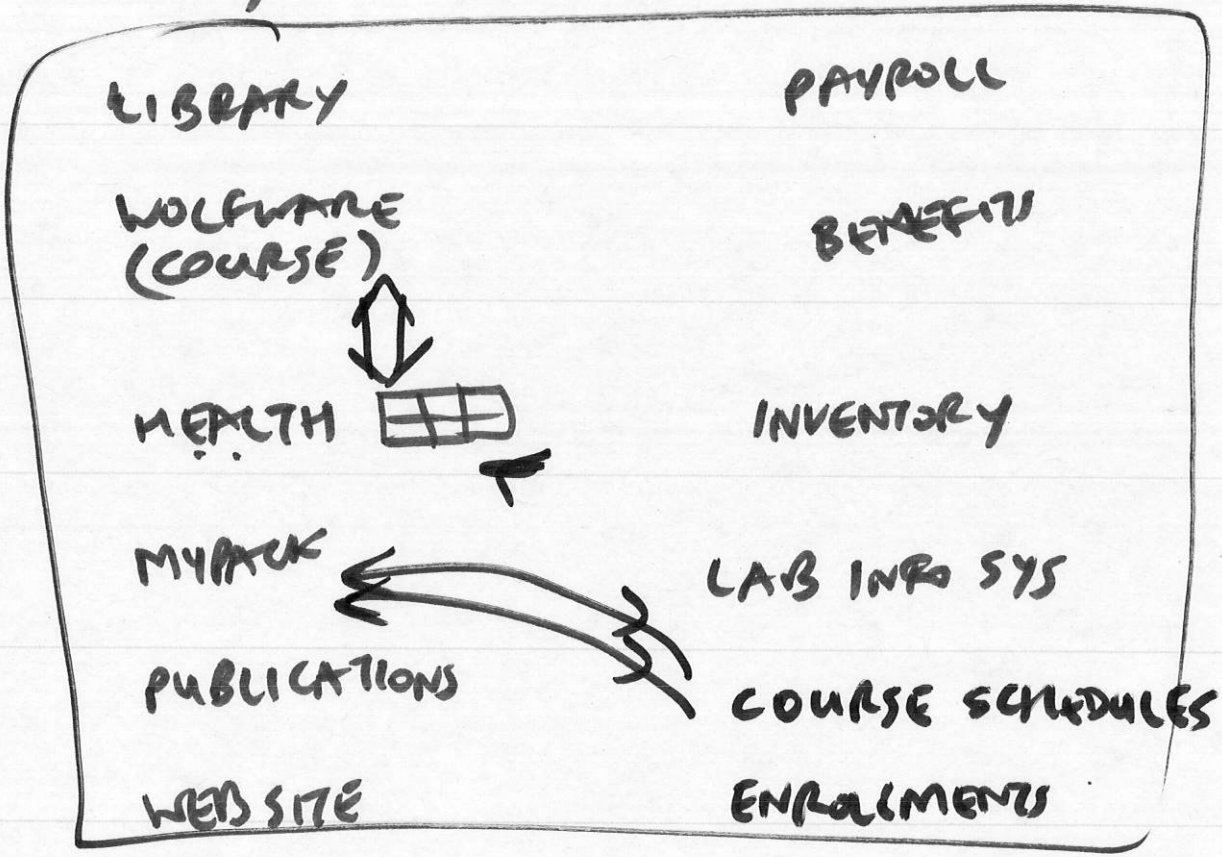
Yields with one integrated entity

- Yields central decision making by homogeneous entity
- Requires resolving all potential inconsistencies ahead of time
- Fragile and must be repeated whenever components change

Obsolete way of thinking: tries to achieve consistency (and fails)

ENTERPRISE : NCSU

KINDS OF INFO RESOURCES (EXAMPLES)



CORRECTNESS CRITERIA (FOR THE INFO IN THE SYSTEM)

DATA VIEW

⇒ CONSISTENCY

COORDINATED UPDATES

REDUCE NEED FOR COORDINATION

ADOPT A PROCESS VIEW

FAILURES

- LOST UPDATE
- INTEGRITY CONSTRAINTS

AUTH

COMMITMENTS

Locality and Interaction

MANY ENTITIES

A way to maintain coherence in the face of openness

- Have each local entity look after its own
  - Minimize dependence on others
  - Continually have interested parties verify the components of the state that apply to them
- Approach: replace global constraints with protocols for interaction (TO THE EXTENT POSSIBLE)
  - *Lazy:* obtain global knowledge as needed
  - *Optimistic:* correct rather than prevent violations
  - *Inspectable:* specify rules for when, where, and how to make corrections

INTEGRATION OF COMPONENTS

- MAINTAIN CONSISTENCY AMONG SE. THEM
- RELIABILITY
- IDENTIFY OVERLAPS AND COMBINE COMPONENTS
- ACCESS OF INFO

Interoperation

Ends up with the original number of entities working together

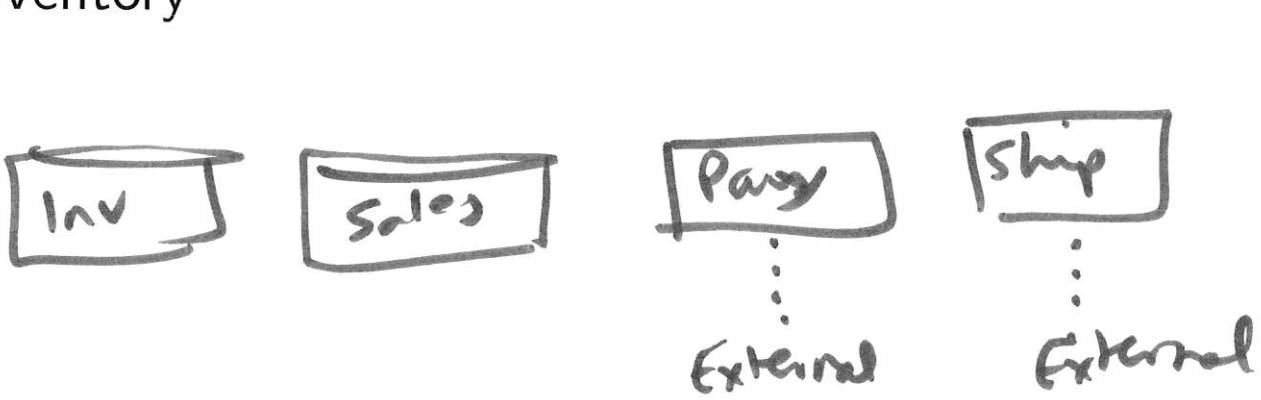
- Yields decentralized decision making by heterogeneous entities
- Resolves inconsistencies incrementally
- Potentially robust and easy to swap out partners as needed

Also termed "light integration" (bad terminology)

Example: Selling

Update inventory, take payment, initiate shipping

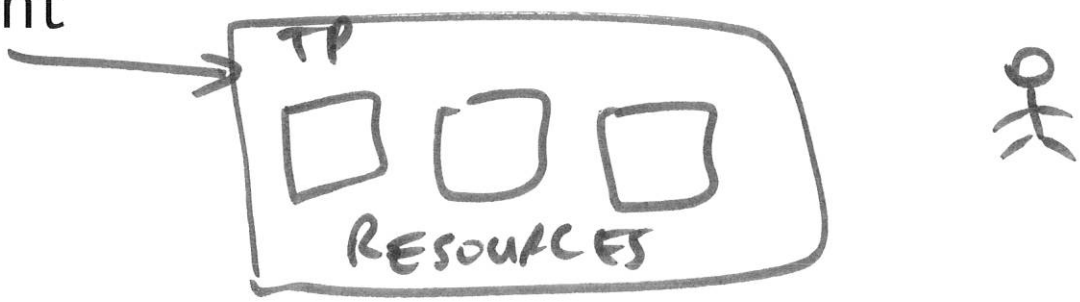
- Record a sale in a sales database
- Debit the credit card (receive payment)
- Send order to shipper
- Receive OK from shipper
- Update inventory



Potential Problems Pertaining to Functionality

- What if the order is shipped, but the payment fails?
- What if the payment succeeds, but the order was never entered or shipped?
- What if the payments are made offline, i.e., significantly delayed?

In a Closed Environment



- Transaction processing (TP) monitors ensure that all or none of the steps are completed, and that systems eventually reach a consistent state
- But what if the user is disconnected right after he clicks on OK? Did order succeed? What if line went dead before acknowledgment arrives? Will the user order again?
- The TP monitor cannot get the user into a consistent state

"IMPOSSIBILITY" OF CONSISTENCY IN AN OPEN SETTING (AND PROGRESS)

HOW WOULD YOU ADDRESS THIS CHALLENGE?

- MAINTAIN STATE
  (LAST GOOD STATE)