Jonathon Duerig

Transactions and Update: A ProtoGENI Perspective

Why Transactions? (Create)

- Allocating resources is cheap
  - Failure is common
- Provisioning them is expensive
  - Failure is rarer

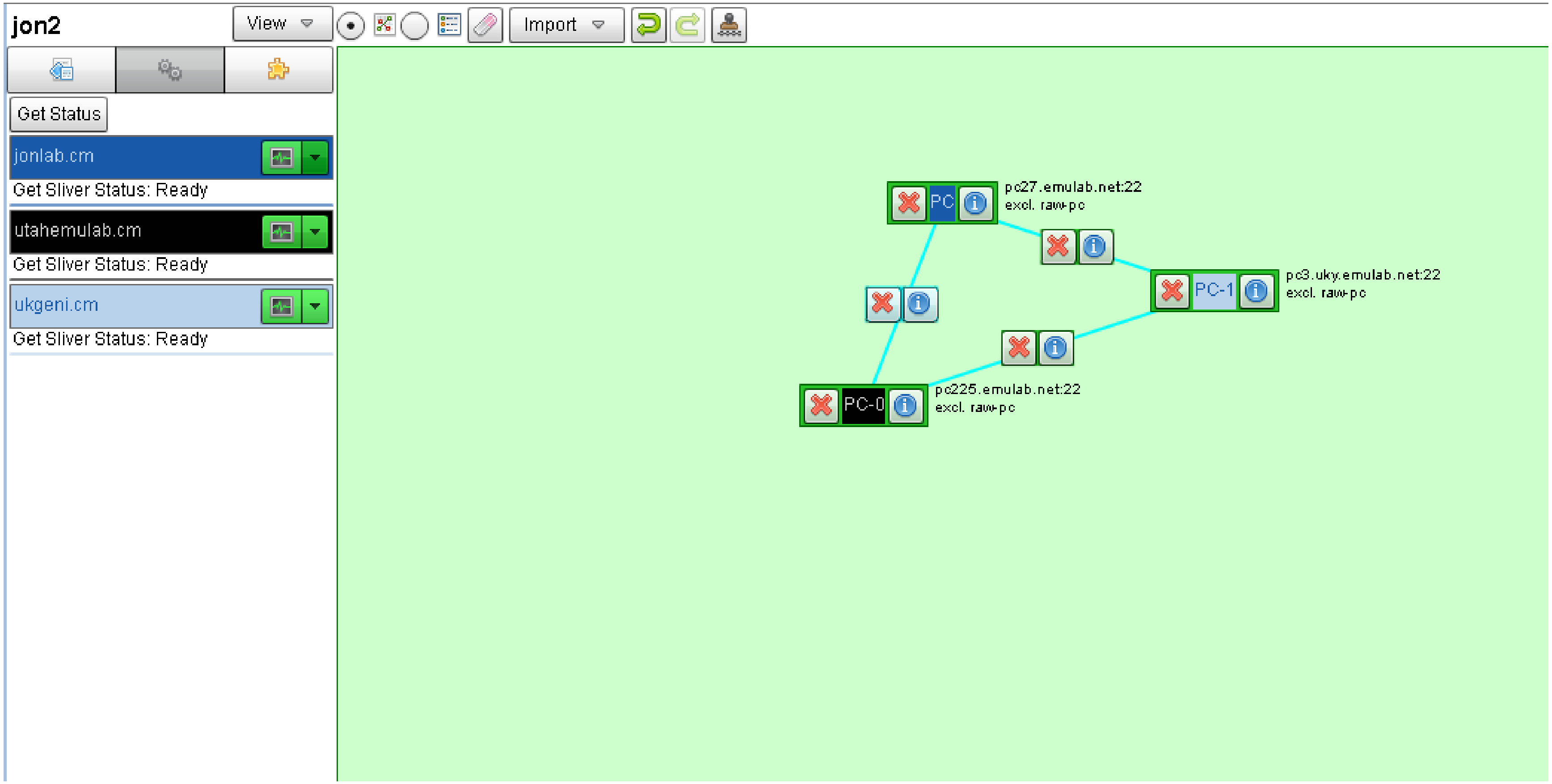
Why Transactions? (Update)

- Update implies
  - Existing Slivers
  - Allocation Change
  - Risk of Loss
- Transactions mitigate risk

Transactional Allocation

- Begin
  - “Guarantee”
  - Short-term Expiration
  - Slivers unchanged
- Commit
  - Slivers are updated
- Abort
  - No loss of sliver state

User Perspective (Update)



jon2

Get Status

jonlab.cm

Get Sliver Status: Ready

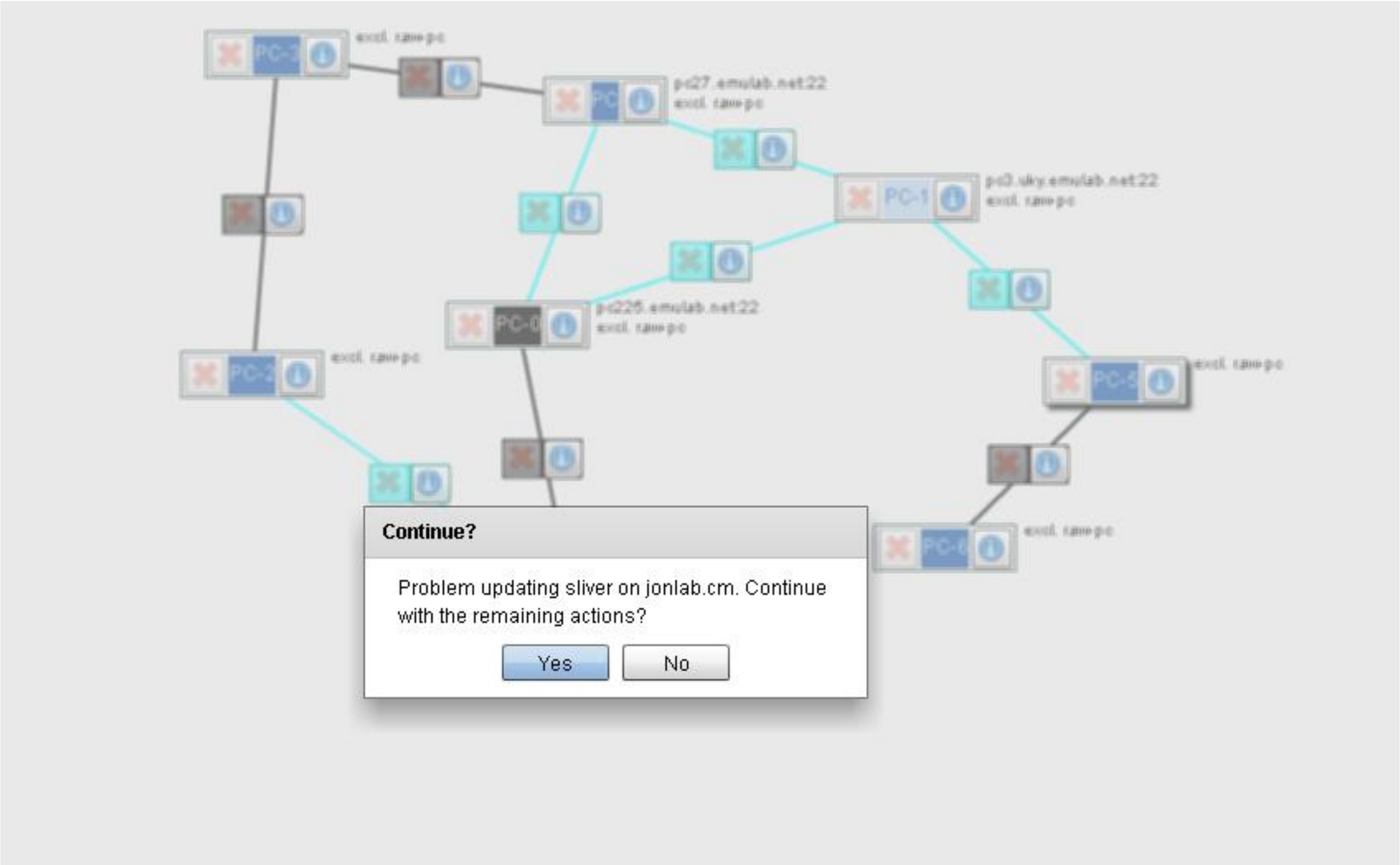
utahemulab.cm

Get Sliver Status: Ready

ukgeni.cm

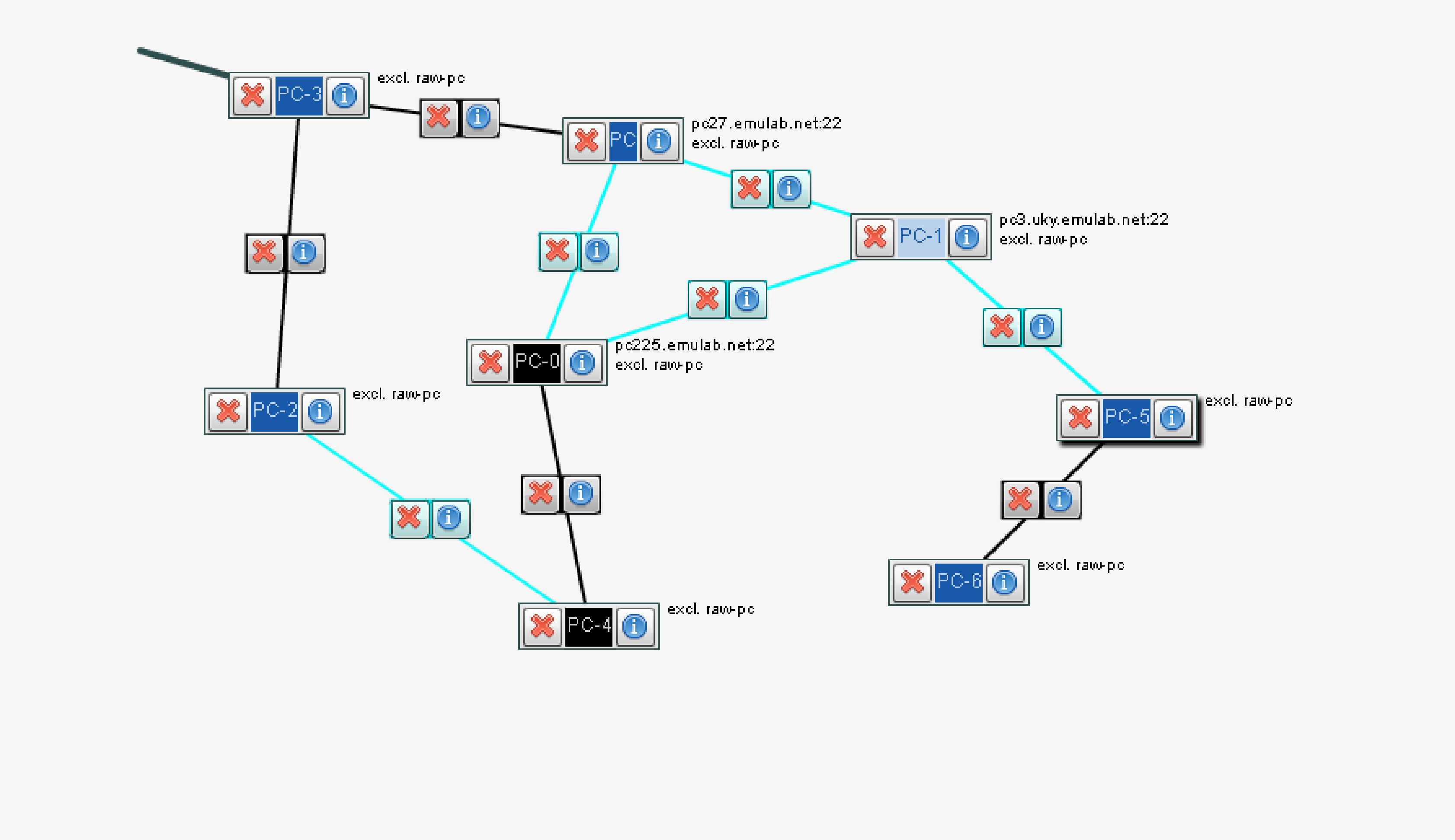
Get Sliver Status: Ready

User Perspective (Problem)



Problem updating sliver on jonlab.cm. Continue with the remaining actions?

User Perspective (Rollback)



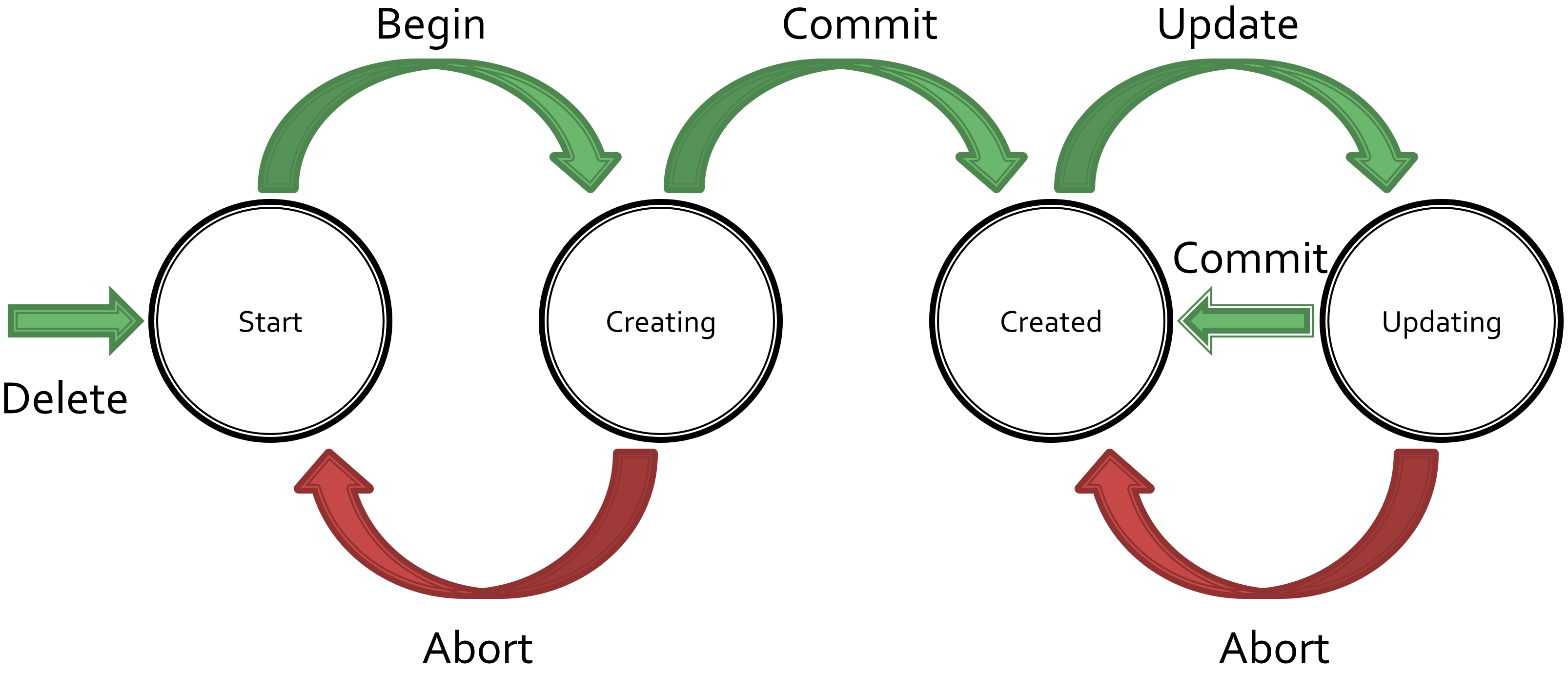
API Goals

- Allow user tools to enforce consistency
  - All or nothing?
  - Take what you get?
  - Change plans?
- Depends on the tool and user

Operation vs. Allocation

- Have I allocated resources at this AM?
  - Allocation state
  - Synchronous
  - Whole aggregate
- What is going on with my nodes?
  - Ground truth about resources
  - Asynchronous
  - Per-resource

API State Machine



Conclusion

- Support Commit/Rollback
  - Especially when updating
  - Need not be tickets
- Whole Aggregate
  - Allocation State Machine
- Per-Resource
  - Operational State Machine
  - Guarantees