ABAC Authorization in GENI
Motivation & Plans

Steve Schwab, Ted Faber

March 15, 2011

www.geni.net

Sponsored by the National Science Foundation

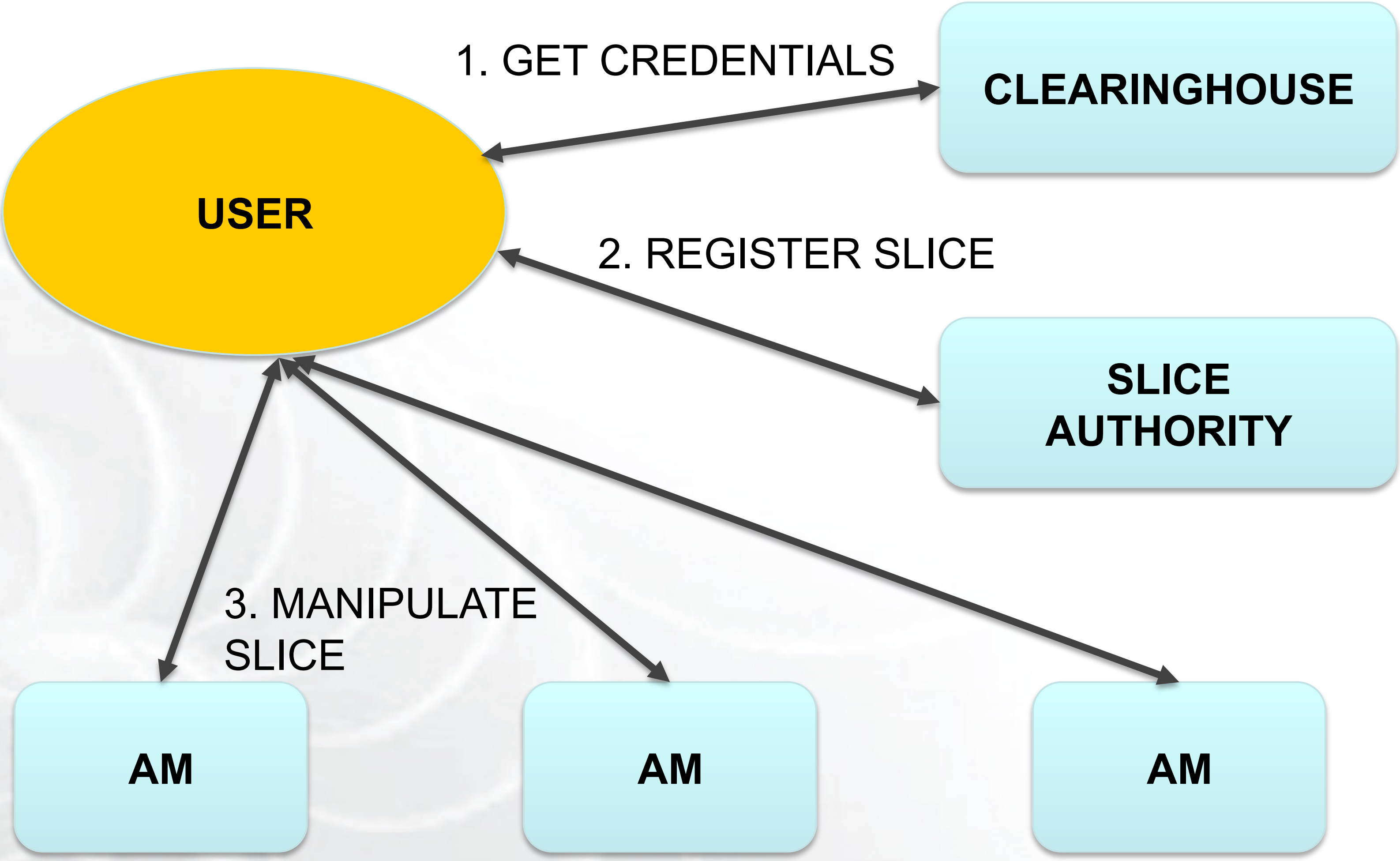
Outline

- Background
  - Authorization Goals
  - What operations does the GENI API allow?
- Motivation
  - Intro to Attribute Based Access Control (ABAC)
  - Approach to using ABAC attribute credentials
  - Current Credentials used in ProtoGENI
- Integration, Development, Trial Deployment Plans
  - GENI API Integration Plan for ProtoGENI
  - ORCA approach
- Summary

Authorization Goals

- Support Many Different Authorization Policies
  - Each control framework/campus/site/research group may want to do things a bit differently.
  - All must enforce some control over who can or can't use their resources.
- Support Many Different Users / Groups of Users
  - Anticipate growth in number of users and distinctions among users.
- Uniform Language for Authorization Policy
  - Cross control framework interpretation of user attributes and resource provider authorization policies
  - Promote sharing and reuse of policies
  - Support auditing

GENI Slice Creation



GENI API: An Operations View

- What does the GENI API allow to be done?
  - SA: register a slice, …
  - AM: request some tickets
  - AM: create_sliver(slice, … , tickets)
  - AM: sliver configuration
- What authorization policy (choices/decisions) must be made? Examples…
  - Can user U create slice S?
  - Can user U allocate a sliver at AM X?

Synopsis of GENI Credentials

- GENI Credentials
  - User X can invoke OP on object O
  - (Subject, Target, Privileges)
    - Types -> Rights -> Operations defines privileges
- Observation
  - This would work great, if we already knew all the types, rights, operations, objects that will ever be needed.
  - And can define intuitively clear names to make policy definition easy.

Sponsored by the National Science Foundation

Motivation for ABAC

- **Start with a well-founded logic and formalism…**
  - A.r1 ← B
  - A.r1 ← B.r2
  - A.r1 ← B.r2.r3
  - A.r1 ← B.r4 ∩ C.r5

  ***Explanation*: users have the rights to do things and delegate things**
- **… hidden inside attribute credentials**
- Decision procedure is precisely defined in a series of papers, with many sophisticated cases worked out… (e.g. information hiding)
  - … don't need most of that power right now, just basic attribute assertions with parameters.
- Avoids locking in to types/rights/operations
  - Can extend as we go… new attributes map one-to-one to new operations or…
  - Portions of the GENI Federation (geographic, control framework, technology type, etc.) can define attributes locally, use them to reduce details in policy definitions.

Architectural Motivation for ABAC

- Shifts future development from “credential format” war to use of ABAC credentials
  - Not important to get a new bit of information into a common GENI AM, ProtoGENI, PlanetLab, or ORCA credential
  - Important to make sure that many parties can generate and interpret ABAC credentials
- Community is free to innovate around ways of using ABAC attributes
  - Relatively de-constraining – attributing and relying parties must agree on meaning of an attribute – and then may adopt them locally for use.

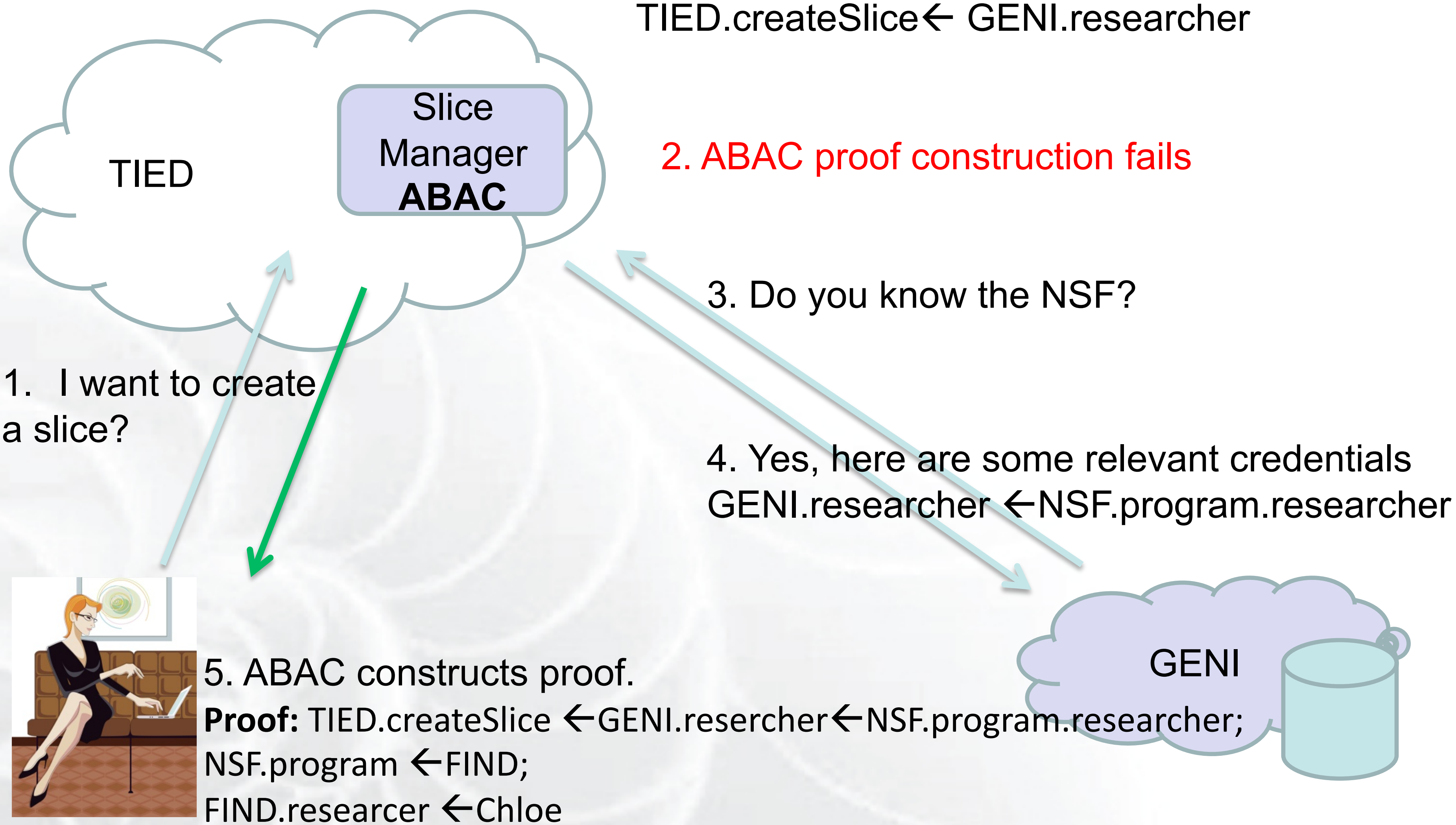
Example of GENI Policy in ABAC Rules

- **The rules for slice creation:**
- AM.slice_authority ← (SA.slice_authority).slice_authority
- AM.slice_authority ← SA
- **The rules for sliver creation are similar to slice creation:**
- AM.ListResources ← (AM.slice_authority).DiscoverResources
- AM.CreateSliver ← (AM.slice_authority).CreateSliver

- The ABAC rules document:
- http://groups.geni.net/geni/attachment/wiki/TIED/ABAC_Rules_v1.2.pdf

- The GENIAPI AM integration document:
- http://groups.geni.net/geni/attachment/wiki/TIED/ABAC_GENIAPIv1.2.pdf

ABAC Enables GENI-wide Authorization Paradigms



TIED.createSlice← GENI.researcher

2. ABAC proof construction fails

3. Do you know the NSF?

1. I want to create a slice?

4. Yes, here are some relevant credentials GENI.researcher ←NSF.program.researcher

5. ABAC constructs proof.
Proof: TIED.createSlice ←GENI.resercher←NSF.program.researcher;
NSF.program ←FIND;
FIND.researcer ←Chloe
Grants Access

ProtoGENI (Near-term) Integration Plans

- **High-level Process**
- Document Design, Implementation and Trial Deployment Plan (Steve, plan document)
- Present Plan & Rough Schedule at GEC 10 (Steve/Ted)
- Get community feedback and consensus at GEC10 (all authors)
- Test and field integration with ProtoGENI and the GPO lab (PG, GPO, ISI)

- The Authorization in GENI plan document:
- http://groups.geni.net/geni/attachment/wiki/GENISecurity/Authorization-plan-v0.4.pdf

ProtoGENI with ABAC Concept of Operations

- Attribute Credentials
  - ProtoGENI to provide ways for users to acquire and pass ABAC Attributes.
- ProtoGENI Reference Policies for Ams
  - AMs may tailor or extend reference policy for local needs.
- Define Vocabulary of Attributes for users and slices
  - Adapt the current ABAC Rules as a starting point, then simplify.
- Enforce ProtoGENI Requestor Semantics
  - Ensure a security check permits only the ID associated with the credentials/ (ABAC assertions) to use those assertions to invoke an operation.
- Discuss long-term plan
  - Dual-credential scheme, or transition to exclusive use of ABAC

ProtoGENI with ABAC

Development, Integration and Trial Deployment

- Analysis Tasks
  - Ensure pieces fit together end-to-end
- Standalone Tools Tasks
  - More tools for ISI to create
- Integration Tasks
  - Within ProtoGENI
  - Within other clients (e.g. Omni)
- Deployment Tasks
  - ProtoGENI and GPO lab
- Field Testing Tasks
  - Recruit potential users

ORCA with ABAC

Compare and Contrast

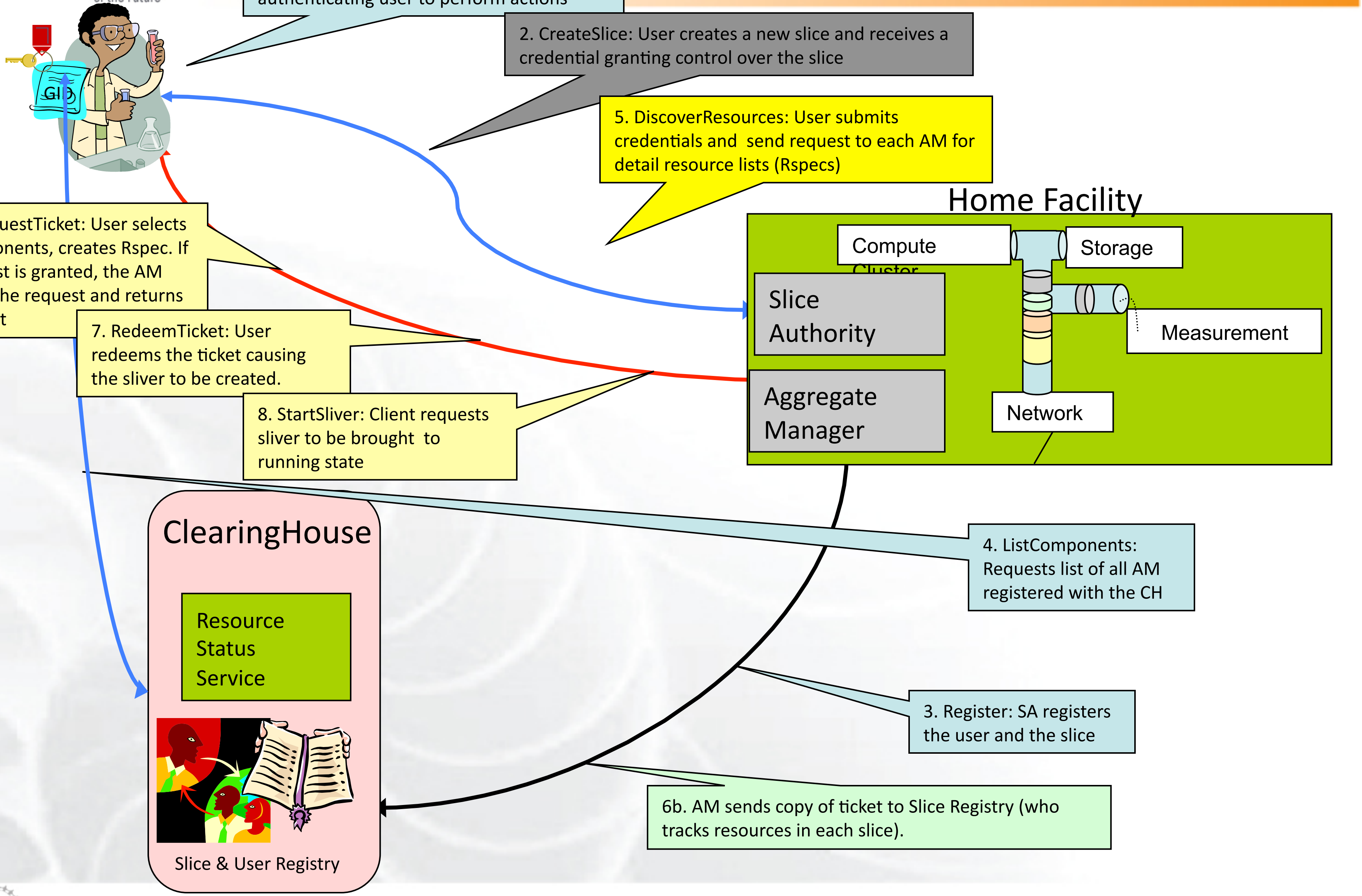
- ORCA mostly in agreement with SFA 2.0 document and GENI API approach
- ORCA contrasts
  - Proposed use of ABAC to encode attributes is less literal, uses attributes such as “Owner” and “SpeaksFor” to introduce a different way of expressing policies
  - AMs are not the only grantor of rights to use resources. Other entities (SA, CH) are anticipated to delegate rights to resources directly to GENI users

Summary

- Current credentials and authorization approach work, but as we make progress in GENI, limitations are starting to creep in.
- Approach provides a chance to experiment with side-by-side implementations of ABAC (assertion) credentials and current credentials
- Most of the necessary software exists
  - Remaining can be developed in next few months
- Policy Definition, Deployment, Use in the Field
  - Experience needed to season prototype, sharpen our collective understanding, reach consensus tipping point.

Backup

GENI Slice Creation



1. GetCredential: S A issues self credential authenticating user to perform actions

2. CreateSlice: User creates a new slice and receives a credential granting control over the slice

3. Register: SA registers the user and the slice

4. ListComponents: Requests list of all AM registered with the CH

5. DiscoverResources: User submits credentials and send request to each AM for detail resource lists (Rspecs)

6. RequestTicket: User selects components, creates Rspec. If request is granted, the AM signs the request and returns a ticket

6b. AM sends copy of ticket to Slice Registry (who tracks resources in each slice).

7. RedeemTicket: User redeems the ticket causing the sliver to be created.

8. StartSliver: Client requests sliver to be brought to running state