Building an EaaSI Network

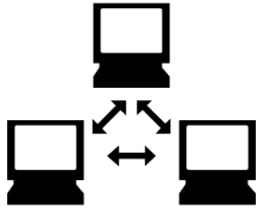
EaaSI Training Module #6

During This Module

- What are EaaSI “nodes” and how do they communicate with each other?
- What is OAI-PMH?
- What happens when I “Publish” an EaaSI resource?

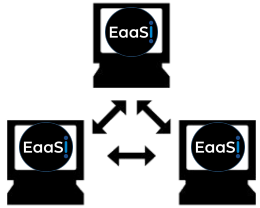
But first...what is a computing "network" anyhow?

Networks

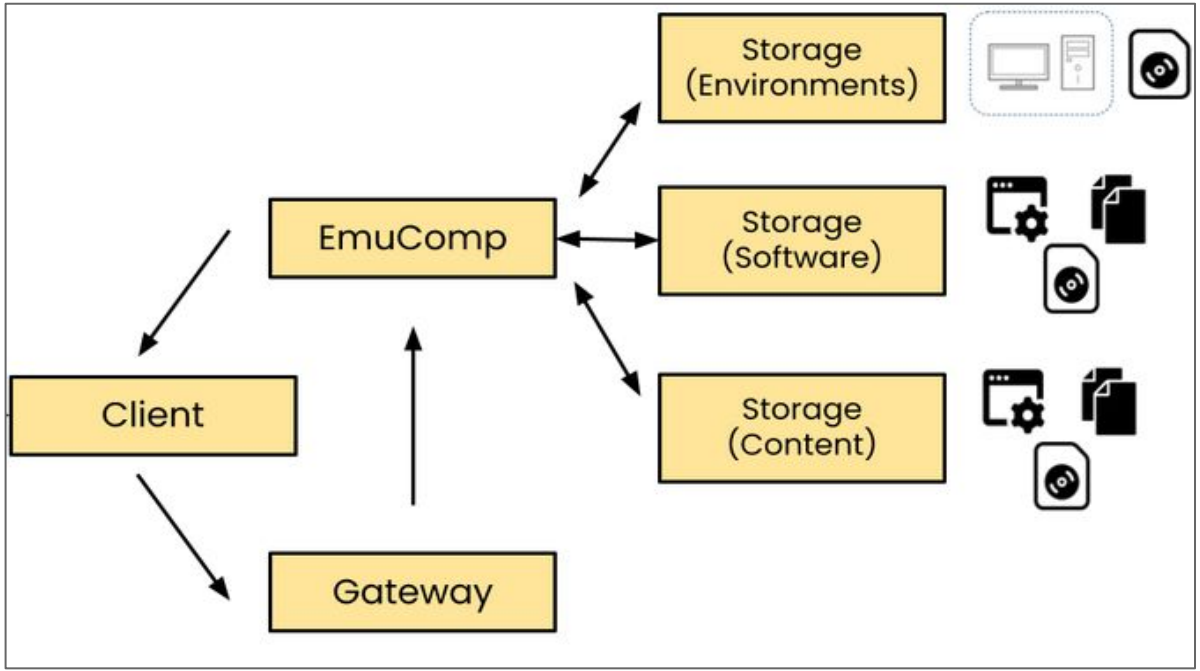


- A set of computers able to communicate and/or share data with each other
- Communicate according to agreed-upon standards and protocols
- Whether it sends or receives information, each individual computer is referred to as a “node”

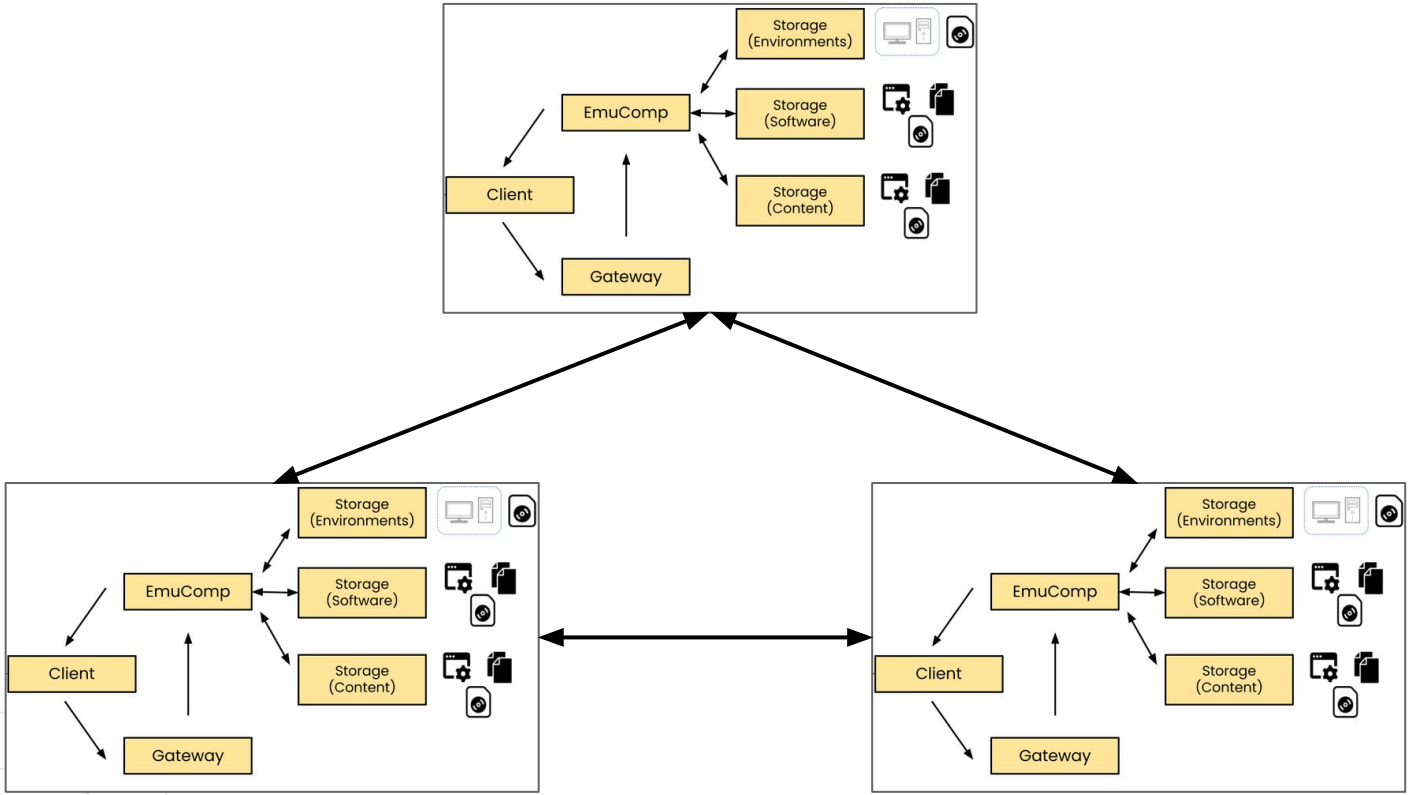
EaaSI Networks



- A set of institutions, organizations, or users able to exchange EaaSI-specific resources with each other (i.e. Environments, Software)
- Communicate over the web, using standard internet and metadata exchange protocols
- One EaaSI “node” = one installation of the EaaSI stack
- Within a node, the organization/institution/user controls their set of resources: which data stays within their node, and which is available to other nodes in the network

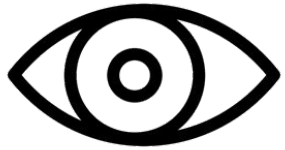


EaaSI "Node" ***(see Training Module #3 for details on the EaaSI software stack):***
https://www.softwarepreservationnetwork.org/eaasi-training-module-3-the-eaas-eaasi-stack



EaaSI "Network" (not precise)

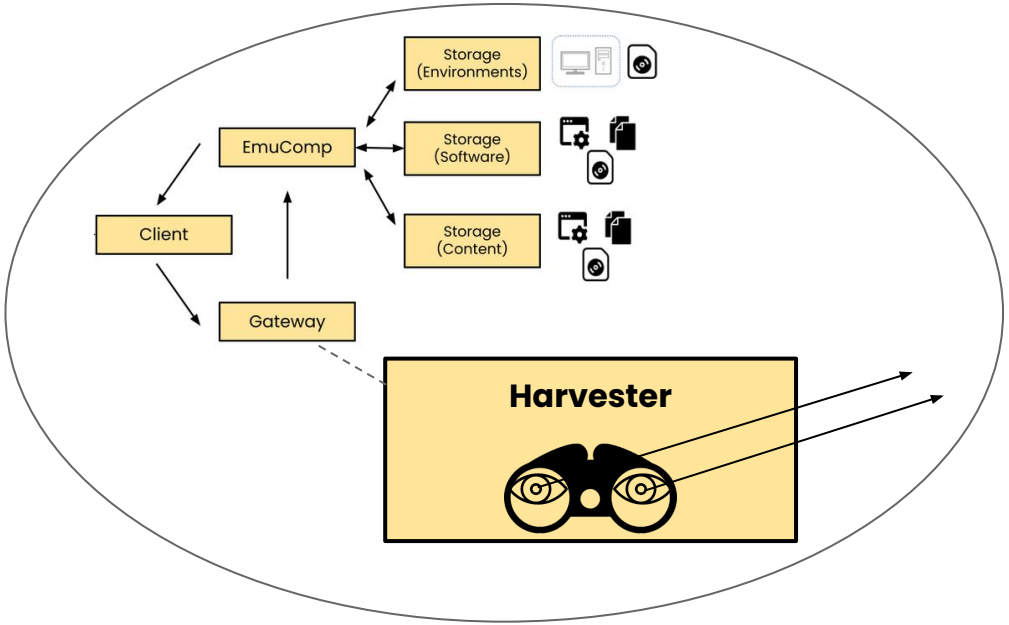
Connecting Nodes: OAI-PMH

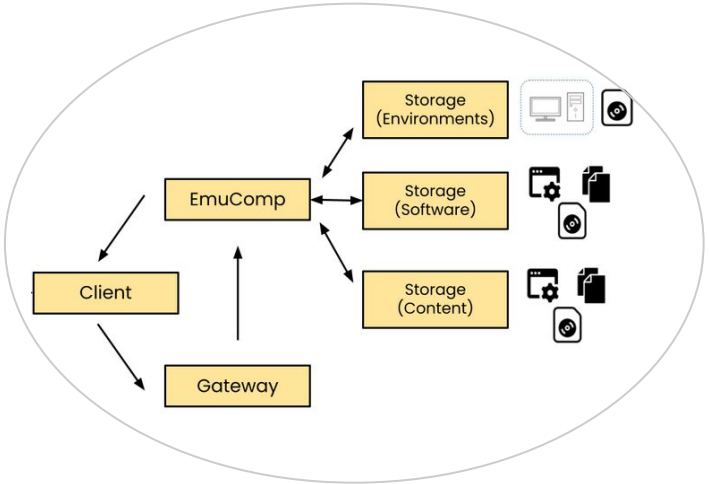


- **O**pen **A**rchives **I**nitiative **P**rotocol for **M**etadata **H**arvesting
- Specifies a mechanism and roles for digital repositories to exchange metadata
  - *Data Providers* are repositories that expose structured metadata for requests
  - *Harvesters* are client applications that request metadata from the Data Provider
- OAI-PMH requests are expressed in HTTP for easy communication over the internet and integration with web apps

EaaSI OAI-PMH

Each EaaSI node stores resource metadata according to OAI-PMH's Data Provider specification, and incorporates a Harvester module to request metadata from other EaaSI nodes



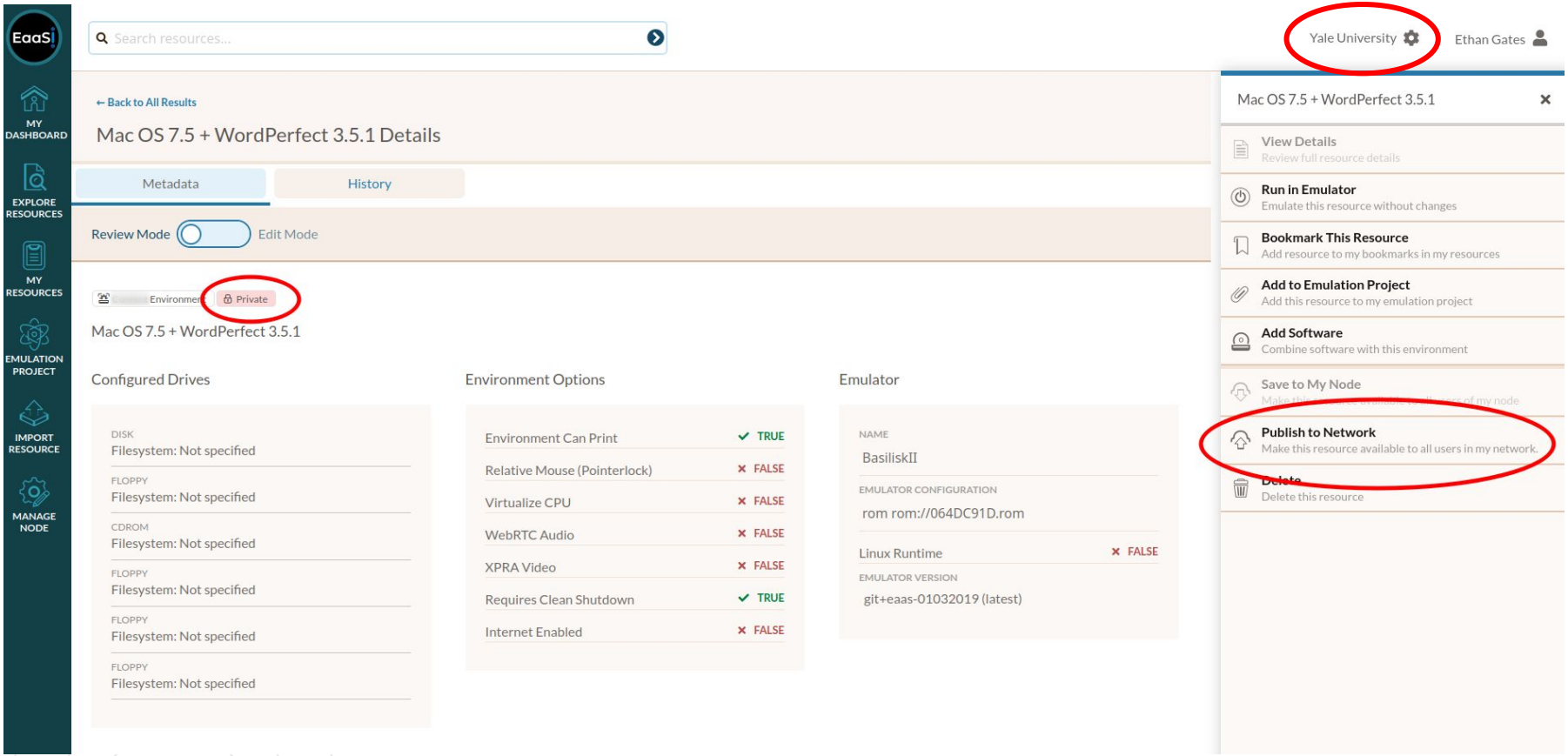


EaaSI OAI-PMH

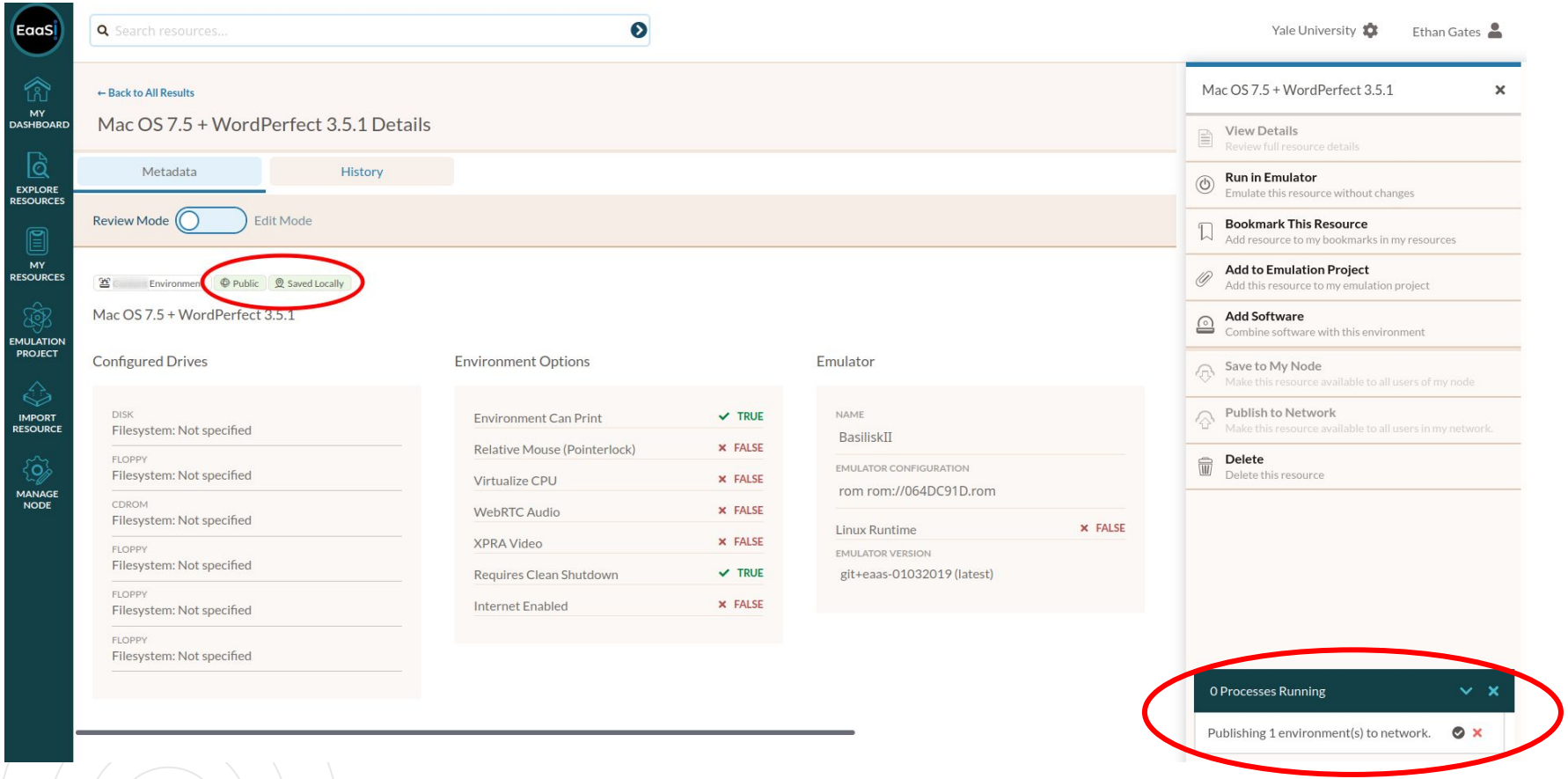
- Resource metadata made available as XML records
- Every resource assigned a UUID (**U**niversally **U**nique **ID**entifier) by EaaSI to make them identifiable across distributed nodes
- Communicates storage location of the resource from the repository/Data Provider, so that other nodes can later find and save that resource to their own storage if desired

Publish

- A decision by the *data provider* node, performed by Admin-level users
- Selecting “Publish” on a resource in the EaaSI UI makes it visible and shareable to other nodes in your network
- Makes resource’s metadata, including all emulation settings, **read-only**
  - Any revision to a published resource will maintain the published resource as-is AND create a new Private resource that incorporates the revisions at the same node with a new UUID



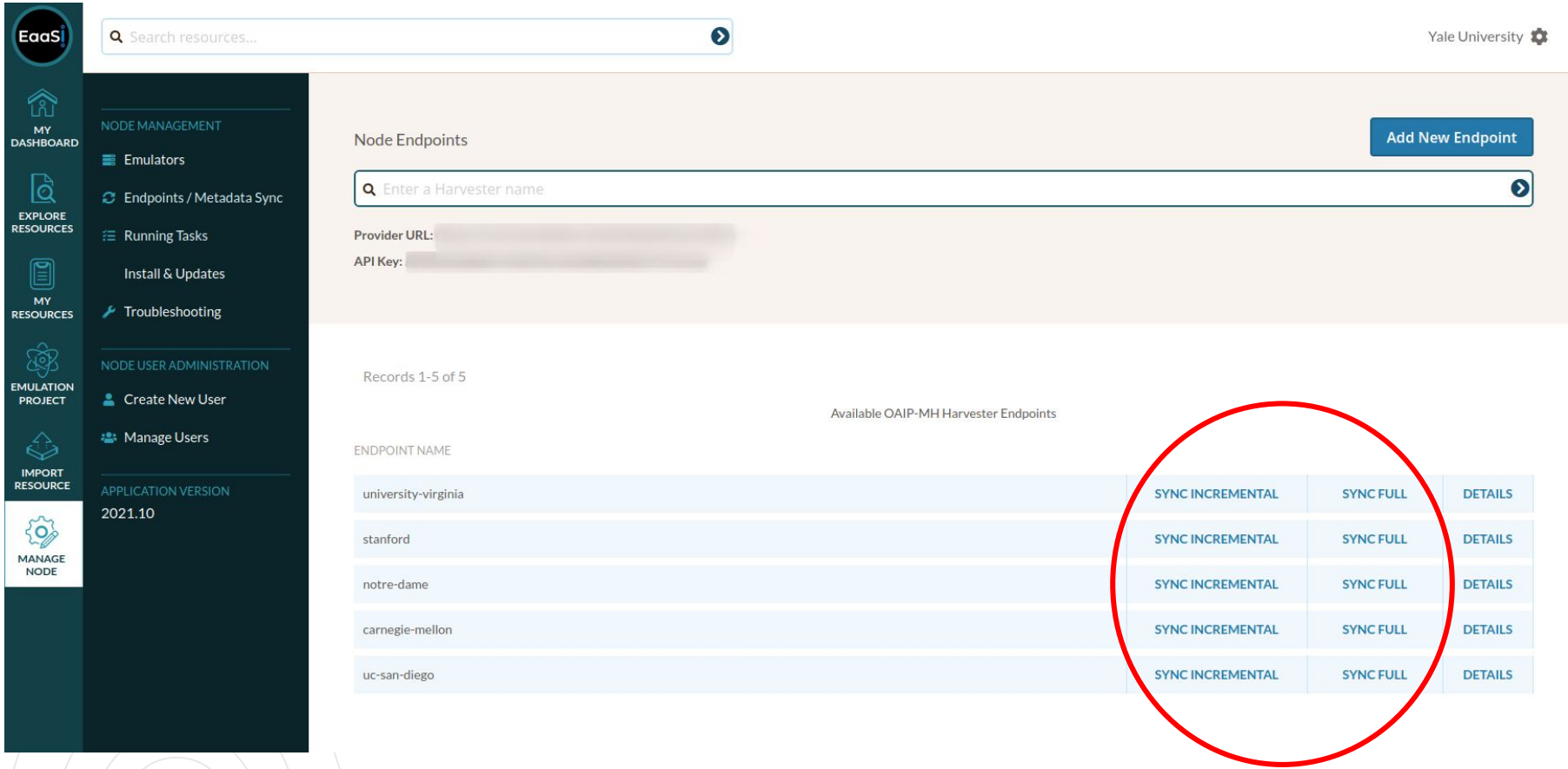
Example: this Environment is only available to this user in the "Yale University" node, as indicated by the "Private" tag. As an Admin, this user has the option to Publish to Network.



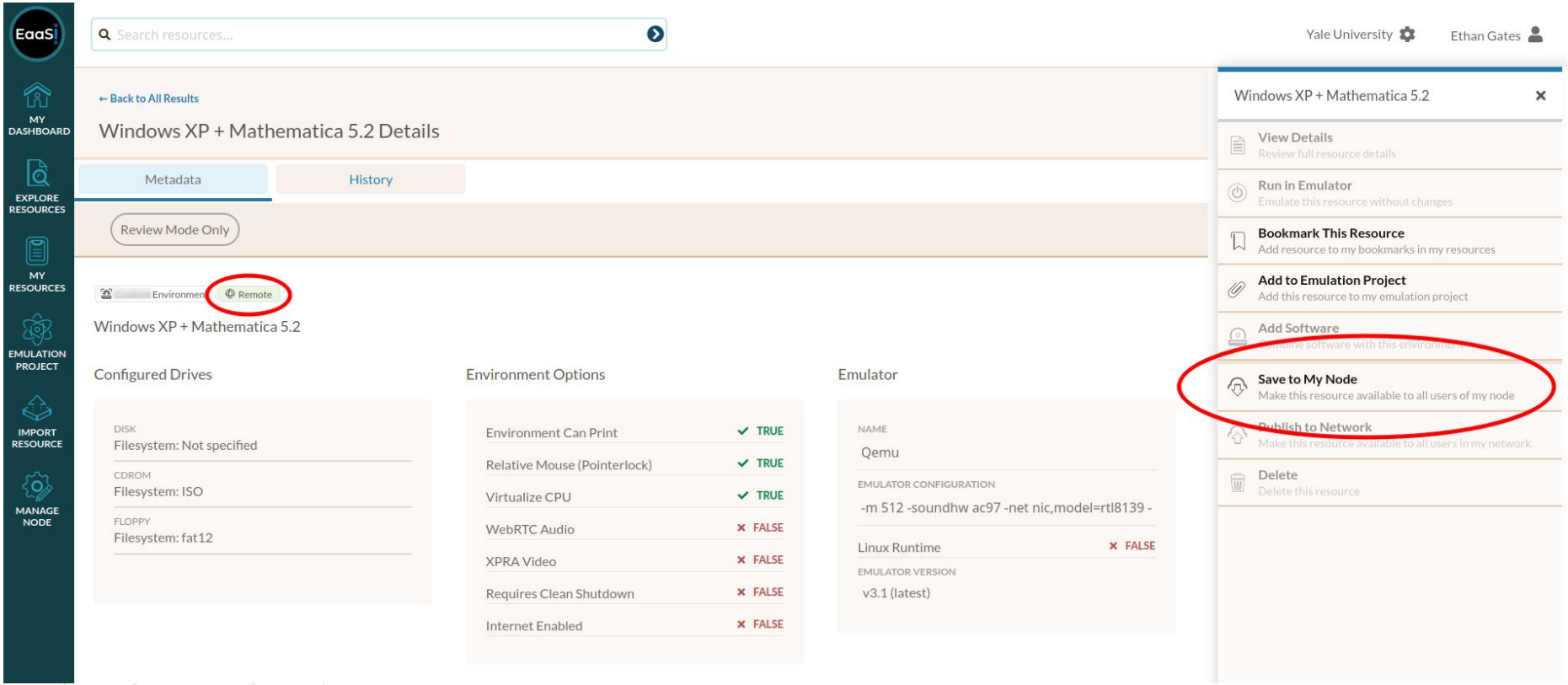
After selecting "Publish to Network," the process completes and the Environment's tags have changed to "Public" (visible and shareable to other nodes) and "Saved Locally" (still available for use within the Yale node).

Sync and Save

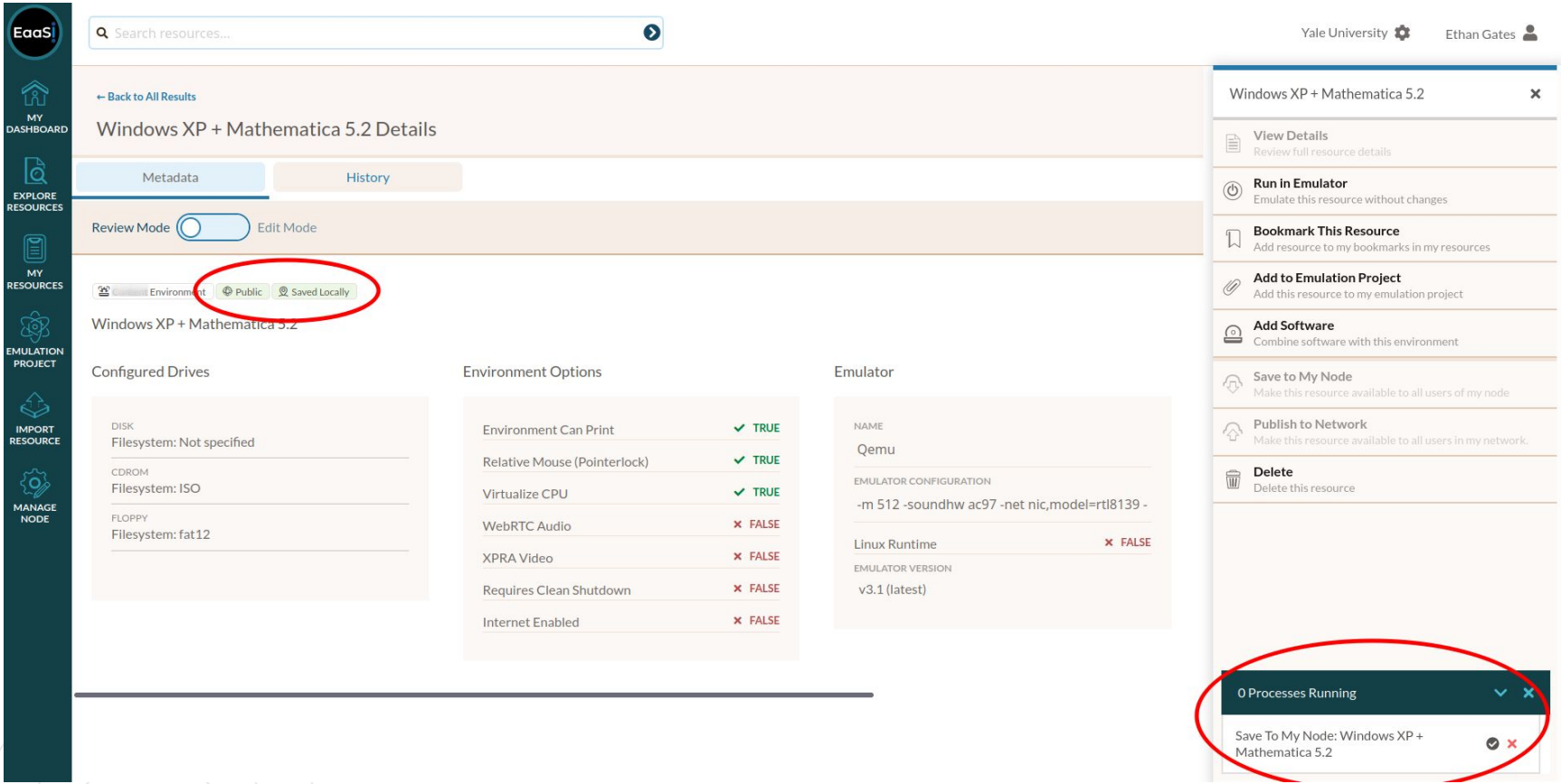
- Decisions by the *harvester* node, performed by Admin-level users
- Sync gathers *metadata* (via OAI-PMH requests)
  - Goal: check what resources have been published remotely at other, specified nodes
- Save gathers *data*
  - Goal: copy the specified resource to your node for local use and storage



Node Admins can set which other nodes to harvest from - and whether to run an "Incremental" Sync (only pick up on changes to available metadata since last sync) or "Full" Sync (completely refresh all that node's available metadata)



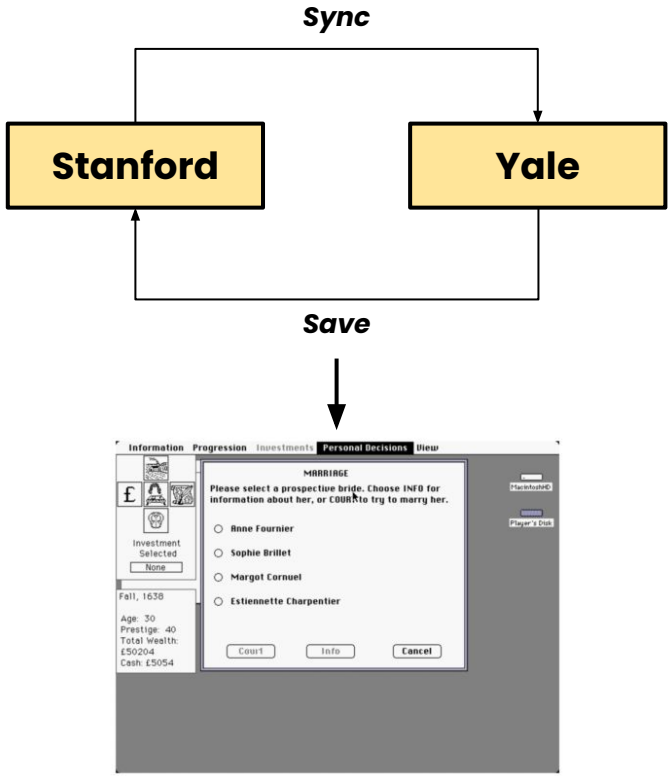
Example: tags indicate this is a "Remote" Environment (available from another node). It can not be "Run in Emulator" - an Admin user must first "Save to My Node."



After selecting "Save to My Node," the process completes and the Environment's tags change to "Public" (still visible and shareable to all nodes) and "Saved Locally" (now copied and can be Run in the Yale node).

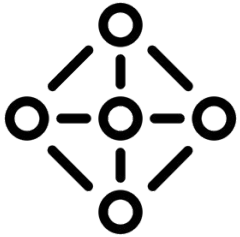
Case Study: “The Would-Be Gentleman”

1. Stanford has a piece of Content (“The Would-Be Gentleman”) written for early Mac OS systems, but no compatible Environment
2. Yale has a compatible Environment - Mac OS 7.01 running on emulated Mac IIci hardware
3. Admin user at Yale publishes Mac OS 7.01 Environment to network
4. Admin user at Stanford syncs to Yale node, saves Mac OS 7.01 Environment
5. Admin user at Stanford uses saved Environment to run Content in emulation



Read the full write-up for more!
https://www.softwarepreservationnetwork.org/eaasi-case-study-1-the-would-be-gentleman/

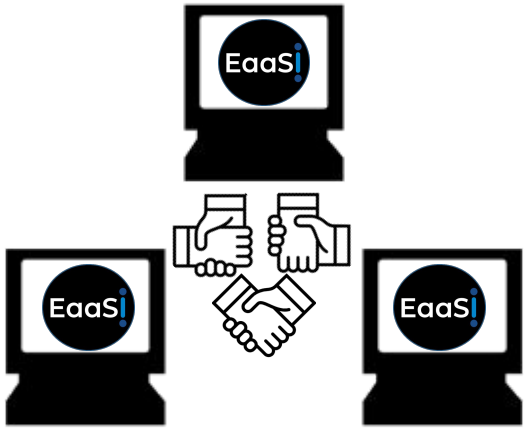
Towards a Distributed Network



- No single node controls all of an EaaSI network's data
- Each node can decide who can harvest from them, and who they harvest from
- Nodes entering or exiting a network does not disrupt the Environments other nodes have created from saved resources
- No obligation to communicate with other nodes; you can use EaaSI without joining an EaaSI network at all, create only Private resources, and never use the stack's OAI-PMH functionality

The Network is the Emulator

- Sharing resources reduces redundant effort
- Collective effort demands collective management
- Requires coordinating standards, schema, acquisition policies



Credits

- Training Module written and designed by Ethan Gates, Software Preservation Analyst, Yale University Library
- All photos, screenshots, and videos recorded by Ethan Gates
- Icons sourced from The Noun Project
- EaaSI program of work sponsored by the Alfred P. Sloan Foundation and the Andrew W. Mellon Foundation, hosted by Yale University Library

Yale
Principle Partner

ALFRED P. SLOAN FOUNDATION
Sponsor

THE ANDREW W. MELLON FOUNDATION
Sponsor