Software-as-a-Service
The iPlant Foundation
Outline

• Why, why, why!
• When ducktape isn’t enough
• Building an API for the web
• Core services
• Love, hate, and building user communities
• Future plans: Roadmap or Apple Maps?
Why, why, why!

iPlant is developing cyberinfrastructure that uniquely enables scientists throughout the diverse fields that comprise plant biology to address Grand Challenges in new ways, to stimulate and facilitate cross-disciplinary research, to promote biology and computer science research interactions, and to train the next generation of scientists on the use of cyberinfrastructure in research and education.
Why, why, why!

Lots of use cases, lots of users.

- Command line
- High performance
- High throughput
- Science gateways
- Big memory
- Big data
- Database driven
- Proprietary codes
- Public codes
- Workflows
- Long-running tasks
- Local hardware
- Web-based
- Long tail science
- Grand challenge
- Interactive
Why, why, why!

Lots of solutions deployed
- Atmosphere: private cloud
- Data Store: virtualized distributed storage
- Discovery Environment: rich web-based science gateway
- TNRS: resolution service to help researcher interact
- Semantic Web: data and service integration based on ontological definitions
- MyPlant: social media site
Why, why, why!

Lots of support given

– Community forums
– Training workshops
– Application porting and optimization
– Collaborative projects
– Pilot programs
– Seed funding

Seed projects: 5
Publications: 70
Collaborators: 140
Workshops: 250
Applications: 420
Users: 7500
Why, why, why!

- Despite all this, there was a clear need for access to the breadth of the iPlant cyberinfrastructure at an API layer.
When ducktape isn’t enough
When duct tape isn’t enough

• Great if solving for X or Y or Z
• Our users needed solutions for X & Y & Z
• Patching things together would actually make the end product more complicated
• Still doesn’t address the missing pieces
When ducktape isn’t enough

- Auth
- Accounting
- Sharing
- Collaboration
- Events
- Notifications
- Namespace
- Provenance
- Common interfaces
Building an API for the web

We started with a simple set of requirements from the community:

– Restful services
– Friendly URLs
– Abstract as many details as possible
– Should work the way the web works
– Encourage good citizenship
– Add value on top of existing services
– Comprehensive, not complete coverage
Building an API for the web

“We are more alike than we are different”

– HTTP
– SSL
– Basic Auth

– JSON
– XML
– Webhooks
Building an API for the web

Setting expectations

– Web vs batch environments
– *big data* vs **BIG DATA**
– Shared environments
– Caching
– Transient systems
– Uncontrolled infrastructure
Building an API for the web

Meeting expectations

– Asynchronous and Synchronous endpoints
– Callbacks & notifications
– Multiple queues
– Preemptive monitoring
– Caching
– Elastic scaling
Core services

Production Services
- Auth
- Profile
- IO
- Data
- Apps
- Jobs
- Postit

Alpha Services
- Audit
- Systems
- Monitor
- Env
- Event
- Mashup
- Meta
Foundation API

https://foundation.iplantcollaborative.org/auth-v1/

Auth Service

/auth/
GET
Validates the username and password/token used to authenticate.

/auth-v1/
PUT
Renews the token used to authenticate for 2 hours.

/auth-v1/
DELETE
Expires the token used to authenticate.
Foundation API
https://foundation.iplantcollaborative.org/auth-v1/

Auth Service

/auth/
POST
Issues a token for a user. The tokens issued can be used interchangeably by the auth service and the rest of the Foundation API services just like actual user passwords.

/auth/list
GET
Returns JSON array with all active tokens for the authenticated user.
Foundation API
https://foundation.iplantcollaborative.org/auth-v1/

Auth Service Future

- Move towards full OAuth2 compliance
- Support delegated authentication decisions
Foundation API
https://foundation.iplantcollaborative.org/profile-v1/

Profile Service

```
/profile/
GET
Returns JSON description of the authenticated user profile

/profile/username/<username>
GET
Returns JSON description of the user with the given username

/profile/search/username/<username>
GET
Returns JSON array containing description of users with matching usernames
```
Foundation API

https://foundation.iplantcollaborative.org/profile-v1/

Profile Service

/profile/search/name/<name>
GET
Returns JSON array containing descriptions of the users matching name

/profile/search/email/<email>
GET
Returns JSON array containing description of users with matching email addresses.
Foundation API
https://foundation.iplantcollaborative.org/profile-v1/

Profile Service Future

– Move towards OpenID compliance
– Support CRUD operations
– Support sandboxed identity management
Foundation API
https://foundation.iplantcollaborative.org/io-v1/

IO Service

/io/list/<username>/<path>
GET
Returns JSON array containing description of the file/folder referenced by <path>

/io/<username>/<path>
GET, POST, DELETE, PUT
Performs operations on files and folders

/io/share/<username>/<path>
GET, POST
Performs operations on file and folder share permissions
Foundation API
Foundation API

https://foundation.iplantcollaborative.org/io-v1/

IO Service Future

– Separate out generic staging functionality into a separate service
– Improve partial data queries
– Support user-defined systems
Foundation API
https://foundation.iplantcollaborative.org/io-v1/

Data Service

/data/transforms

GET
Returns JSON list of all available transforms

/data/transforms/<username>/<path>

GET
Returns JSON list of available transforms for a specific file

/data/transform/<transform_name>

GET
Returns JSON description of the transform
Foundation API
https://foundation.iplantcollaborative.org/io-v1/

Data Service

/getdata/tag/<tag_name>/

GET
Returns JSON list of transforms tagged with the given term.

/postdata/async/transform/<transform_name>/<username>/<path>/

POST
Returns the file transformed from its original format to the named format and staged to a location defined by the user.

/getdata/sync/transform/<transform_name>/<username>/<path>/

GET
Returns the file transformed into the named format.
Foundation API

https://foundation.iplantcollaborative.org/io-v1/

Data Service Future

– Improve partial data queries
Foundation API
https://foundation.iplantcollaborative.org/apps-v1/

Apps Service

/apps
POST
Add or update an app you own by posting an app description file

/apps/list
GET
Returns JSON array containing a list of all public apps

/apps/shares/list
GET
Returns JSON array containing a list of all your public and shared apps
Foundation API

https://foundation.iplantcollaborative.org/apps-v1/

Apps Service

GET, POST, PUT, DELETE
Manage the specified app.

GET
Search for applications by the given field and value

GET
Returns a JSON array of applications
Foundation API

https://foundation.iplantcollaborative.org/apps-v1/

Apps Service

GET

/apps/shared/<app_id>

Search for private and shared applications by the given field and value

GET

/apps/shared/[name|tag|term]/<value>

Returns a JSON array of the user’s private and shared apps

GET

/apps/shared/list

Returns a list of private and shared apps.
Foundation API

https://foundation.iplantcollaborative.org/apps-v1/

Apps Service

/apps/<app_id>/share
GET, POST, DELETE
Manages ACL for the given application

/apps/<app_id>/form
GET
Generates a HTML form that can be used to submit a job for the given app

/apps/
POST
Register, update an application
Foundation API
https://foundation.iplantcollaborative.org/io-v1/

Apps Service Future

– Support cloning of apps.
– Make app publication easier.
– Support different application execution types
– Support different execution platforms
– Better support for individual environment setups.
# Foundation API

https://foundation.iplantcollaborative.org/apps-v1/

## Systems Service

<table>
<thead>
<tr>
<th>Path</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/systems/list</td>
<td>GET</td>
<td>Returns JSON array containing descriptions of all available iPlant execution systems</td>
</tr>
<tr>
<td>/systems/list/&lt;resource_id&gt;</td>
<td>GET</td>
<td>Returns JSON description of the given system. Resource_id is XSEDED IIS ResourceID</td>
</tr>
</tbody>
</table>
Foundation API
https://foundation.iplantcollaborative.org/io-v1/

Systems Service Future

– Better monitoring
– POST method for updating
– Support registering of private systems
– Support convenience endpoints for status, etc.
– KBase integration
Foundation API
https://foundation.iplantcollaborative.org/monitor-v1/

Monitor Service

GET
/monitor/<yyyy-mm-dd>
Return JSON array of daily monitoring results

GET
/monitor/service/<service_id>/<yyyy-mm-dd>
Returns JSON array of daily monitoring results for a service

GET
/monitor/suite/<service_id>/<yyyy-mm-dd>
Returns JSON array of daily monitoring results for a suite

GET
/monitor/test/<test_id>
Returns JSON representation of a specific test
Foundation API
https://foundation.iplantcollaborative.org/io-v1/

Monitor Service Future

– Deeper testing of individual systems
– Add networking tests
– Ramp up testing frequency when possible
– Add application and data movement tests
Foundation API
https://foundation.iplantcollaborative.org/apps-v1/

Job Service

/job
POST
Submit a job request

/job/<job_id>
GET, POST, DELETE
Query for information, resubmit, or delete a job

/job/<job_id>/share
GET, POST, DELETE
Manage job share permissions
Foundation API
https://foundation.iplantcollaborative.org/apps-v1/

Job Service

/job/<job_id>/input
GET
Returns JSON array containing descriptions of the inputs for the job

/job/<job_id>/output/list/<path>
GET
Returns JSON array of job output files (similar to IO, but unique to the job)

/job/<job_id>/output/<path>
GET, POST
Downloads the file at the given path relative to the job’s output folder
Foundation API
https://foundation.iplantcollaborative.org/apps-v1/

Job Service

/job/list/[<attribute>/<value>]+

GET
Searches for jobs by one or more attribute value combinations

/job/<job_id>/<attribute>

GET
Returns JSON array of jobs with only the job id and specified attribute
Foundation API

https://foundation.iplantcollaborative.org/apps-v1/

Job Service

/jobs/list
GET
Returns JSON array containing user jobs
Foundation API
https://foundation.iplantcollaborative.org/io-v1/

Job Service Future
- Add support for OSG, multi-cloud, FutureGrid, and user-defined systems
- Support CLI apps
- Support submissions run under individual user accounts
- Support arbitrary archiving locations
- Improve rate limiting, provenance, and data movement efficiencies
- Implement a charging models
- KBase integration
Foundation API
https://foundation.iplantcollaborative.org/postit-v1/

Postit Service

/postit/
GET
Creates a disposable url with built in authentication. Returns JSON array containing postit url and lifetime information

/postit/<nonce>
GET
Invokes the url registered to this postit with the registered method. Any form data passed in to this PostIt is forwarded, as appropriate, to the registered target.

/postit/<nonce>
DELETE
Revokes the specified PostIt immediately.
Foundation API
https://foundation.iplantcollaborative.org/io-v1/

PostIt Service Future

– Expand as more general authenticated url shortening service.
Foundation API
https://foundation.iplantcollaborative.org/audit-v1/

Audit Service

/audit/
GET
Returns JSON array containing job, iPlant Data Store, and Atmosphere usage information.

/audit/<job,data,atmo>
GET
Returns JSON object containing the specified form of usage information.
Foundation API

https://foundation.iplantcollaborative.org/audit-v1/

Audit Service Future

– Add better support for Atmosphere accounting
– Add data movement analytics
Foundation API
https://foundation.iplantcollaborative.org/event-v1/

Event Service

/event/
GET, POST
Register a new event or get a snapshot of all your registered events

/event/<event_id>
GET, POST, DELETE
Manage an event

/event/<event_id>/subscriptions
GET, POST
List and create subscriptions.
Foundation API
https://foundation.iplantcollaborative.org/event-v1/

Event Service

```
/event/<event_id>/subscriptions/<sub_id>
```

GET, POST, DELETE
Manage an existing subscription.
Foundation API
https://foundation.iplantcollaborative.org/event-v1/

Event Service Future

- Integrate directly with underlying messaging services
- Incorporate global IDs into registration requests
- Integrate API events as first class events in service
Still to come…

Meta Service

/meta/
POST
Create a new metadata item

/meta/<meta_id>
GET, POST, DELETE
Manage metadata item

/meta/schema
GET, POST
Returns a JSON array of schemas registered with the Service. Create a new schema.
Still to come...

Meta Service

/meta/schema/<schema_id>
GET, POST, DELETE
Manage a metadata schema

/meta/map
POST
Returns a JSON Array of metadata schema maps registered with the service. Create a new metadata map between schemas.

/meta/map/<meta_id>
GET, POST, DELETE
Manage a metadata schema
Still to come…

Meta Service Future

– Improve scalability
– Explore better ways to translate between schemas
– Directly support ontology overlays
– Integrate tightly with the other API services
Still to come…

Mashup Service

/mashup-v1/
POST
Submit a mashup for execution

/mashup-v1/<mashup_id>
GET, POST, DELETE
Returns JSON descriptions of a specific mashup

/mashup-v1/list
GET
Returns a list of JSON descriptions of your mashups
Still to come...

Mashup Service

/mashup/tags/<tag>
GET
Search for mashups by tag

/mashup/share/<mashup_id>
GET, POST, DELETE
Manage share permissions on a specific mashup

/mashup/share/<mashup_id>
POST, DELETE
Performs operations on job share permissions
Still to come…

Env Service

– Still thinking through this service
– Primarily needed when we support individual system registration.
– Will handle user environment management
– Update, query user environment
– Full module support
– Discovery endpoints
– Migration endpoints
Love, hate, and building user communities
Love, hate, and building user communities

Foundation API Usage by Month

Cumulative Monthly Requests

Month of Usage

Systems
Profile
Monitor
Jobs
IO
Data
Auth
Apps

THE UNIVERSITY OF TEXAS AT AUSTIN
TEXAS ADVANCED COMPUTING CENTER
Love, hate, and building user communities
Love, hate, and building user communities
Things we’ve done well (thus far)

• Listened to our users
• Responded to user feedback,
• Implemented bug fixes, and feature requests quickly and openly
• Listened to our users
• Handled early scalability issues
• Documented the API, the concepts, and the process
• Kept our services running through down times
• Been preemptive about issues and updates
Teaching Points

• Address versioning early
• Be obsessive-compulsive about regression testing.
• Communicate with your users early and often.
• When dependencies go down, you look bad, so design with failovers in mind.
• Big data doesn’t move well on the web. Set user expectations early and often. Don’t be afraid to get creative and provide alternative solutions.
Teaching Points

- No one else thinks it’s as simple as you do.
- The success is in the science.
- Take the time to onboard developers.
Future plans: Roadmap or Apple Maps?

• Don’t build stuff just to build stuff
• Don’t aim at nothing, you’ll hit it every time
• Listen to our users. They’re the only reason we’re here.
Future plans: Roadmap or Apple Maps?

2.0 features release
- Global identifiers
- New HPC, HTC, Cloud, and private systems
- New application types
- 100’s of new apps
- Performance improvements
- Stability improvements
Future plans: Roadmap or Apple Maps?

New services

– OAuth2: fully compliant with current spec
– Staging service: from anywhere to anywhere using any protocol
– Metadata: deep metadata support for the entire enterprise
– Mashup: store and share workflows
Developer’s Console

https://foundation.iplantcollaborative.org/
Demo App
https://foundation.iplantcollaborative.org/iplant-test
More Info

http://iplantcollaborative.org

https://foundation.iplantcollaborative.org/docs

https://foundation.iplantcollaborative.org/forums
For more information:
www.iplantcollaborative.org