
2i2c Team Compass

2i2c

Mar 16, 2022

TEAM PRACTICES

1	How to use the team compass	3
2	Team Practices	5
3	Managed JupyterHub Service	19
4	Team Reference	23
5	About 2i2c	41
	Index	51

This Team Compass is a guide for team members of 2i2c to navigate our organization and community. It is the *source of truth* for information about 2i2c.

HOW TO USE THE TEAM COMPASS

2i2c team members should document everything relevant to operating the organization in the 2i2c team compass, and give it precedence over all other kinds of documentation unless explicitly stated otherwise in the team compass.

The Team Compass is the source of truth!

Any information that is in this team compass should be considered the **source of truth** for 2i2c. Its information should be complete and updated frequently.

If you see information here that is out of date, please propose an edit in [the team compass repository](#).

See below for a brief overview of each section of the Team Compass, or browse the sections to the left on your own.

See also:

For help navigating the team compass, see *[Navigating the team compass](#)*.

TEAM PRACTICES

Information about working with the 2i2c team, understanding our processes and projects, and generally getting oriented.

2.1 Get started at 2i2c

This is a guide and checklist for new team members of 2i2c to get oriented, and to make sure that you're set up with all the resources you need to start contributing!

A note about staff vs. non-staff

The information here (and in the Team Compass) applies to anybody that is “on the 2i2c team”. That means both staff at ICSI, as well as non-staff team members that may work in other institutions. However, there may be some steps here that are only relevant to ICSI staff.

2.1.1 Navigating the team compass

This repository acts as a knowledge base and a “Source of Truth” for everything at 2i2c. It is regularly updated as our team processes, structure, and information changes. There is a *lot* of information in the team compass, so here is a quick primer on where to look next.

To **get started with the 2i2c Team**, see the sections below:

- *Where information is located* - an overview of where different kinds of information is stored for 2i2c (e.g., what things are stored in Google Drive, what things in GitHub, etc)
- *Expectations of team members* - what practices you are expected to abide by while working with the 2i2c team.
- *How we communicate* - the various channels that the 2i2c teams use for communication, and what kinds of conversation happens in each channel.
- *How we coordinate and plan* - the team workflow that we use in carrying out projects and daily activities.
- *How to update the team compass* - how to update the team compass itself.

In addition, the other sections on this page contain pointers to other useful sources of information for 2i2c.

2.1.2 Onboarding process

New 2i2c team members are onboarded in order to make sure that the following conditions are met:

- They know where to look for certain information
- They feel welcome and excited to be a part of the team
- They have an understanding of what 2i2c is currently up to
- They have enough information to get started

When a new team member joins, follow these two steps first:

1. Identify an **Onboarding Champion**. This is somebody that will help guide the onboarding process for our new team member. Their job is to carry out the process described in the onboarding issue template.
2. [Open an issue in the Team Compass](#). Choose the “Onboard a new team member” template. This issue will track the onboarding process, and will provide steps to take in order to onboard our new team member.

The following sections describe some things that should be met in order to onboard somebody new.

2.1.3 Accounts that should be created

- **A GMail address**. You can use this to access information in the 2i2c drive, as well as various online services that we use such as cloud engines. You can sign in to your email address at gmail.com (it behaves like a Google account).
- **A Slack account**. In addition to GMail, 2i2c also uses Slack to have informal or synchronous communication.

2.1.4 Accounts you should have access to

- **Google Drive**. Your 2i2c email address will get you access to a Google Drive that contains documents, brainstorms, and other assets that aren't fit for the 2i2c Team Compass. You can log into the Google Drive by going to drive.google.com and adding a new account using your @2i2c.org address and password.
- **GitHub Access**. 2i2c has a few GitHub repositories. You should have membership in the [2i2c-org](https://github.com/2i2c-org) GitHub organization and write access to its repositories. Check that this is the case and, if not, ask for access in the Slack.
- **NameCheap Access**. 2i2c runs much of its cloud infrastructure at custom URLs created for the target institution. In order to ensure you have the ability to create new URLs, you should have access to the 2i2c NameCheap account. If not, ask for access in the Slack.
- **Cloud Accounts**. 2i2c has a number of cloud accounts that it uses to create and operate its infrastructure. You should have access to the cloud accounts for any projects that you expect to work on. This includes Google Cloud, Microsoft Azure, and Amazon Web Services. The specific project might depend on what you're working on, so check with the team to make sure you have the right access.

2.1.5 What are we working on right now?

The *2i2c team coordination page* has information about our daily workflow. You can use this to begin participating in our daily activities and find out what everyone is working on.

2.1.6 What major projects is 2i2c working on?

Check out the *projects page* for more information about the projects that 2i2c is working on.

2.1.7 Tips for distributed working

2i2c is a distributed organization, and believes strongly in following practices that are inclusive, participatory, and collaborative. It has team members spread out over many time zones working on a variety of projects. There are many guides and tips for distributed working^[^distributed-work1], and we've tried to distill a few key components for our workflows:

1. **Have a single source of truth.** For any information or projects in 2i2c, there should be one source of truth. Any conflicting information will defer to this source of truth, and it should be updated first and often. The default source of truth is this team compass, unless otherwise specified.
2. **Document everything.** Documentation is the most important tool for coordinating and distributing information across distributed teams. It is crucial that 2i2c team members document all relevant information about their projects, what they are working on, etc.
3. **Broadcast updates in multiple places.** Different people have different preferred methods of engagement and communication on 2i2c projects. For this reason, don't assume that posting an update or question in a single location will reach everybody that you wish to reach. Do not hesitate to post questions or ask for feedback in multiple places (e.g., GitHub, Slack, or even email). However, try to keep information in a single place to have a single source of truth.
4. **Update often.** Err on the side of updating the team more often than not. Nobody should be surprised when something gets done - especially nothing major. Your work should be as transparent to others as possible!

2.2 Where information is located

2.2.1 General organizational information / documents / brainstorm

We also have some organizational documents in a shared Google folder called "2i2c". Your 2i2c address will give you access to this folder (or if you do not have access, please request it!):

https://drive.google.com/drive/folders/1ABxxSFycGfCzQc9czfwer_dat-GVi4jw?usp=sharing

The folder is organized into a few top-level folders that *should* be relatively self-explanatory, but don't hesitate to ask for a quick guide.

2.2.2 Todos, project plans and management

2i2c uses *the Projects page* to keep track of the major projects that it is working on. That page has links to the location of project-specific to-do lists, deliverables, etc.

More organization-specific discussion and items we track can also be found in *the private 2i2c meta repository*.

Generally speaking the daily specifics of activity on a project are recorded *outside* of the team compass, as they are more dynamic. The important thing is that the team compass make it clear *where* to find this information, and that the information is kept up-to-date for all.

2.2.3 Marketing and communication

There are three main ways that 2i2c communicates with the outside world:

- Twitter: [@2i2c_org](#)
 - Currently, there is nobody actively monitoring the Twitter account. We use it to post links to blog posts or major announcements in the newsletter.
- Newsletter: [using Mailchimp](#)
 - Users sign up to the newsletter via our landing page’s contact form at [2i2c.org#contact](#). Email addresses are imported into Mailchimp for the newsletter.
- Blog: [\[2i2c.org/blog\]](#)
 - The blog is managed by our [Hugo website repository](#). All new posts are automatically sent to the mailing list on a weekly basis.

2.2.4 Slack

Who is invited to the Slack?

Currently, anyone who is interested can join the 2i2c Slack. Initially this is people that mentioned they would like to join via our blog post. Any Slack member is welcome to send an invite link to another person that would like to join.

Private vs. Public rooms in the Slack

There are a mix of private and public rooms in the Slack. In general, conversations about projects, development, etc should be in public rooms. **Most conversation in the 2i2c Slack should be in public channels.**

We’ll create private rooms for a few specific topics that probably warrant private conversation. By default, we’ll start with:

- [#partnerships](#) - to discuss new partnerships with 2i2c
- [#recruiting](#) - to discuss hiring with 2i2c

and eventually

- [#sales](#) - to discuss new sales leads etc

Conversation about more sensitive topics like *pursuing* collaborations, sales, etc should be in a private room until it’s appropriate to start discussing in public channels. By default, all 2i2c staff, key volunteers, and founders will have access to all private rooms (it is only the “friends of 2i2c” and users that do not have access).

Code of Conduct

2i2c is still in the process of creating its Code of Conduct, so it does not have a 2i2c-specific CoC to apply to the Slack. In the meantime, all members of the 2i2c Slack must abide by the [NumFocus Code of Conduct](#) guidelines. If you wish to report a violation of this code of conduct in the 2i2c Slack, send an email to conduct@2i2c.org.

Warning: This section will be updated once 2i2c defines its own CoC and reporting infrastructure.

What rooms to use?

We try to keep the number of rooms to a minimum, and only add new rooms if it really feels necessary (e.g. if we keep having “off topic” conversations about the same topic in one room). Over time, we can revisit the Slack rooms that we use, and add / prune as needed.

Bots

We use the [GitHub bots](#) to track github issues that are relevant to Slack channels, and explore more integrations in the future. In general we should only use the bot to mirror comments if a repository is *obviously always relevant* to a channel. Otherwise, only mirror comments that are *tagged with a particular label* (e.g. [#recruiting](#)).

2.3 Expectations of team members

This page describes some of the expectations for 2i2c team members. While these are not strict rules, they are things you should do in order to help maximize its impact.

2.3.1 General team practices

Be respectful

When interacting with 2i2c team members, or individuals outside of 2i2c, always be respectful. We are currently working on a Code of Conduct for 2i2c, but until it is complete, we expect everyone to follow the principles described in the [Jupyter Project Code of Conduct](#).

Be communicative

First and foremost, be communicative to other team members. 2i2c is a distributed organization, and it is crucial that we are open and up-front with one another. If there’s any doubt, don’t hesitate to ask someone, ideally in a public channel. If you are wondering whether somebody else needs to know something, assume that they do and tell them.

Be proactive

We are a distributed organization, and this means that it is crucial that we work together to identify how to make the most impact. 2i2c generally do a combination of development, operations, and support for the infrastructure and projects that we work on.

What you work on will depend on your role and potentially your funding source, but the important thing is that you are proactive in working with other team members to identify what you should work on next.

Read *How we coordinate and plan* and the other documentation in this guide to understand how the 2i2c team coordinates amongst one another.

2.3.2 Engineering Team Practices

These team practices are relevant for the engineering team at 2i2c.

Participate in the team workflow

The 2i2c Engineering Team has *a team coordination workflow here*. You are expected to participate in this workflow with other members of the team.

Participate in the following activities

If you're an engineer at 2i2c, plan to do a combination of these things:

1. **Development for the *Managed JupyterHub Service*.** This is a collection of JupyterHubs that we run for various customers/communities in research/education. We are constantly updating and improving this infrastructure!
2. **Operation for the *Managed JupyterHub Service*.** In addition to developing the infrastructure, we also operate it and provide support. We try to divide this work so that we share the responsibility of operation and support.
3. **Focused development in collaboration research and education communities.** In addition to running infrastructure, we also collaborate around specific projects, usually involving tools in the Jupyter ecosystem. You may work with researchers in improving their tool, or building something new for another group's project.
4. **Outreach, teaching, and community-building.** In addition to technical work, we also encourage 2i2c team members to make interactive computing more accessible and powerful for others via outreach, particularly for under-served communities.
5. **Open source development and support.** We run all of our infrastructure on open source tools - usually from communities for which we are not the sole leader. It is crucial that we participate in these communities and support them in addition to doing 2i2c-specific work.

2.4 How we communicate

There are a few different communication channels with 2i2c, depending on your team and what kind of communication you'd like to have. See below for details.

2.4.1 Asynchronous communication

Most communication should be asynchronous - this means you communicate without the expectation that another person immediately responds. Asynchronous communication is more inclusive of team members that are spread across many time zones and with many responsibilities. By default, try to keep it async!

General discussion between the team

For those who are doing daily work with 2i2c, we use [GitHub Discussions](#) for our asynchronous conversation. There are a few different discussion forums depending on the kind of conversation you'd like to have:

- General team discussion: [2i2c-org/team-compass/discussions](#)
- Discussion specifically about the pilot hubs: [2i2c-org/pilot-hubs/discussions](#)
- Team discussion that needs to be private (this should be low-volume): [2i2c-org/meta/discussions](#)

To-do items and deliverables

For tracking actionable items and conversation around them, 2i2c uses GitHub issues and pull-requests. Check the repositories in the [2i2c-org GitHub org](#) for where these conversations happen.

In addition, check [the active projects page](#) for a list of active projects and links to where communication is happening around them.

Team members are expected to monitor repositories that are important to the work that they are doing. Keep up-to-date with new issues and comments on pre-existing issues.

Daily workflow planning

See the [team coordination page](#) for information about how we communicate around our priorities and daily workflow.

Steering Council discussion

The Steering Council of 2i2c conducts asynchronous discussion via [GitHub Issues in the meta/ repository](#). However, this may change in the future ([see this issue for context](#)).

2.4.2 Synchronous communication

Synchronous communication assumes that there are one or more people communicating directly with each other at the same time. This is usually for informal communication between team members (ie, information that does not need to be tracked over time).

Important: Any conversations that are important or that should be remembered outside of the immediate context should be encoded as Discussion threads, GitHub Issues, or otherwise put in a more asynchronous place so that others may discover it and participate.

Here are things that **should not** be conducted synchronously, but instead put in an issue or other “Source of Truth” location:

- Important updates on the status of a project or issue
- Decisions that were made from conversations in Slack

- Important dates or deadlines announced on Slack
 - Major bugs or problems that were uncovered on Slack
 - New ideas that warrant follow-up
-

Team coordination

Team coordination generally happens via the 2i2c Slack channel (<http://2i2c.slack.com/>). In particular in the `#team-updates` channel as well as in the hub-specific channels (e.g., those beginning with `#hub-`). This is generally used for quick coordination, hand-offs, and requests for help or reviews. It is not used for official records or planning.

Informal communication

Informal communication happens in the 2i2c Slack channel (<http://2i2c.slack.com/>). This Slack has both 2i2c staff and Steering Council members, as well as several other interested parties.

2.5 How we coordinate and plan

As a totally distributed team, it is important to have team practices that give one another insight into what we're working on, identify opportunities to collaborate, and un-block one another. This page describes some of the practices that we adopt towards these goals.

2.5.1 What are our goals?

Here are the goals we optimize for in organizing our team coordination practices.

- Give team members insight into what one another is working on (not to “report up”)
- Look for opportunities to work together and/or support one another's work efficiently
- Provide some social accountability to get stuff done

We want to do this with high efficiency so that people can quickly get on the same page and focus efforts around decision-making and collaboration. We also wish to do it in a remote-friendly way since we are split across many different time-zones.

2.5.2 Team development workflow

This section describes how our development team carries out its planning and day-to-day work. Here's a quick summary:

- *Deliverables* are discrete units of value that we bring to others. To be ready for work, they should be scoped properly, with enough information to complete them (e.g., a set of *tasks*).
- The *Project Backlog* contains lists of **deliverables** that we are currently working towards across all of the 2i2c projects. Deliverables on the Project Backlog define our team priorities at any moment.
- *Tasks* are concrete actions to take, and should be accomplishable in a single work session by one person.
- The *Activity Backlog* contains lists of **tasks** that we are currently working on. This defines 2i2c's current activity.

Project Backlogs

The Project Backlog contains all of the deliverables across our projects that we wish to work on quickly. These are organized into a few columns, representing the *state* of each deliverable. Here is a common column structure:

- *Needs Discussion/Refinement*: Deliverables that are high-priority but un-refined. Our goal should be having discussion and doing research in order to get these deliverables ready for work.
- *Ready to Work*: Deliverables that are well-scoped and have a clear path forward, and are thus ready to implement. As deliverables in *In progress* are completed, we should replace them with deliverables from this column. Generally speaking, deliverables near the top have higher priority than those at the bottom.
- *In progress*: Deliverables that we are currently working towards. This means that we should be generating *Tasks* in our *Activity Backlog* to complete this deliverable (more on this below).
- *Blocked/Waiting*: Deliverables for which we are waiting for some action that is out of our team's immediate control.
- *Done*: Deliverables that have been completed. We should close these issues and celebrate the improvements that we have made!

Note: In general, there should be only two or three deliverables per team member, per column on this board. It should not become so heavily-populated that it is hard to keep track of deliverables!

There are two Project Backlogs at 2i2c:

- The *2i2c Development Projects Backlog* contains deliverables for our development-focused projects.
- The *Organizational Project Backlog* contains deliverables that are organization-wide and not related to development.

Deliverables

Deliverables represent incremental amounts of value we can deliver to a particular stakeholder, and should be completable in a week or two.

Most issues in our repositories are deliverables, in varying states of readiness. When a deliverable is first created, it may lack information, be improperly scoped, or have an unclear path to implementation. We improve this through *deliverable refinement* (see below).

A deliverable is ready to work (and can thus be added to the Project Backlog) when it has the following properties (adapted from the *INVEST methodology*).

- **Have a short title and description:** Deliverables should be glanceable and have enough context that the reader can quickly understand the scope.
- **Have one or more user stories:** User stories should define who benefits from the deliverable, and why they want it.
- **Have completion criteria:** We have clear completion criteria for this deliverable to denote it as “done”.
- **Have tasks to complete it:** Deliverables should have a set of tasks, which are actions needed to complete the deliverable.

How refinement happens

Deliverable Refinement is the process of improving the scoping, context, and structure of our Deliverables issues so that they are ready for us to work on them. When a deliverable is created, it may not have all of the information needed to take the next step. Adding that information is the goal of Deliverable Refinement.

All team members are expected to participate in deliverable refinement, though the more experienced and higher-level you are, the more you should be contributing to this process. The important thing is that we always have a list of high-quality deliverables ready to work towards.

Here's a small table that explains how to decide whether deliverables issues should be put in a project backlog:

	Needs more information	Ready for working
Low priority	Stays in issues	Put in <i>Ready to work</i>
High Priority <i>Needs Discussion/Refinement</i>	Manually put in Put in <i>Ready to work</i>	

Activity Backlog

The **Activity Backlog** contains the collection of **Tasks** that the team is currently working on. Tasks are actions that are needed to accomplish some deliverable. These tasks are generated from the deliverables on the Projects Backlog. They define a “to do” list of tasks to complete on a day-to-day basis.

The Activity Backlog is broken down into these columns:

- *Needs Discussion/Refinement* Tasks that require some team discussion around *implementation*. If there are higher-level discussions about the deliverable itself, perhaps the deliverable is not yet ready to be worked on.
- *Ready to Work* Tasks that are ready to be worked on. Roughly speaking, tasks higher on the list are of higher priority.
- *In progress* A task that a team member is currently working towards. When you move a task to In progress, indicate that you are working on it (either by adding your username to the card, or “assigning yourself” in the issue/PR associated with it).
- *Needs review* Tasks that have an implementation that require feedback from others. For example, reviews for pull requests.
- *Done* Tasks that are complete! When you move a task here, make sure to update any relevant deliverables.

There is one Activity Backlog defined at [this GitHub Projects Board](#).

Tasks

Deliverables of sufficient complexity are broken down into *tasks*. Tasks should be completable by a single person in less than one day (generally, completable with a few hours of focused work at most). Make sure to attach a link between your deliverable and its tasks (e.g., by adding links to issues in the deliverable’s “tasks” checklist)

Tasks are generally encoded as checklist items in a deliverable, then added to the Activity Backlog when the deliverable is being worked on.

Here's an example of a deliverable with a few tasks:

- **Deliverable:** Side notes in Jupyter Books.
User story: *As a book author, I want to be able to create “aside” content that does not break up the narrative flow of my text.*
Tasks:

- Create design doc for directive name and general API
- PR to prototype functionality and documentation
- Add tests and QA
- Final approval + merge

2.5.3 Weekly team syncs

Every Monday, the 2i2c Hub Operations team conducts a team sync to get on the same page and coordinate their tasks for the week.

A GitHub Action will automatically create a new Team Sync issue at the beginning of each Monday. This action uses [this issue template](#) and inserts currently open/actionable issues before creating the sync issue. It uses [this Python/ipynb script to generate the issue](#).

Here is the process for our team syncs.

1. **On Mondays a new issue is created automatically.** This issue is our space to discuss, update, and sync.
2. **Team members give their responses.** You can copy/paste the template, and then give your responses in comments to the issue. You shouldn't feel forced to add content if you can't think of anything, use it as much as is useful.
3. **Discuss and agree on next steps throughout the day.** The goal is to ensure that important issues have somebody paying attention to them, and that team members are supported in the tasks they work towards. At the end of the day the issue is automatically closed.

Note: While these syncs happen once a week, the process of communicating with team members and working on tasks / deliverables can be dynamic and constantly updating. The syncs are just to get everyone on the same page.

2.5.4 Monthly team meetings

The 2i2c Team meets the **first Monday of each month** for two hours. This is an opportunity to speak face-to-face, discuss major issues and work items, and coordinate around our major next steps.

We use [this HackMD](#) for our team meeting notes and agenda.

See the [Team Calendar page](#) for information about when and where the meetings are held.

2.5.5 Taking time off

First and foremost, 2i2c fosters a culture of healthy balance between work and life. All 2i2c team members should take the time they need to thrive - this will both make you happier and help us accomplish our mission most effectively.

The 2i2c team uses a [shared Google Calendar](#) to keep track of when team members will be away. Everyone at 2i2c has access to it, and has the ability to add events to this calendar.

Use the Calendar to let the team know you're taking time off. Here are the reasons that you might do so:

Personal Time Off: If you're taking personal time off (e.g., vacation, holidays, personal leave, etc), add an event to this calendar with a title to let others know you'll be away.

For example:

AWAY: <your name> - <optional reason>

Reduced time: If you expect to be on **reduced time** for an extended period, please use the calendar for this as well.

For example:

REDUCED: <your name> - <optional reason>

Holidays: Please add any national holidays for your location of residence if you expect that you (or others in your location) may take time off for this as well.

For example:

HOLIDAY: <holiday name>

2.6 GitHub repository conventions

This page describes some common conventions and patterns that we follow in our GitHub repositories.

2.6.1 Issue labels

There are a few issue labels that we use to provide key metadata in our issues. These are added to all 2i2c-org/ repositories, and share the same meaning across each.

Note:

- Only **Issue Type** is required for all issues.
 - Repositories may define their own labels in addition to the ones described here for a given category, unless otherwise noted.
-

Issue Type

REQUIRED

Issue type determines the kind of issue and implies what sorts of actions must be taken to close it. Issue types are mutually-exclusive - there may only be one issue type per issue.

There are a few issue types that are defined for all repositories:

- type: enhancement: an improvement to be made to the repository.
- type: bug: something that is not working or incorrect in the repository.
- type: task: an action to take that is not well-categorized by enhancement/bug.

In addition, other repositories may use repository-specific types, with the caveat that **all issues must still only have one type label**.

Issue priority

OPTIONAL

Issue priority is used to classify some issues as requiring action before others. Any issue without a priority tag should be assumed as lower priority than `prio: low`.

Here are the priority labels for our issues:

- `prio: high`
- `prio: medium`
- `prio: low`

Issue tag

OPTIONAL

Tag labels provide hints about what topics that issue is relevant to. They are highly repository-specific, optional, and non-exclusive (so issues may have many tags associated with them).

Here are some example tags:

- `: documentation:` related to documentation in a repository
- `: CI/CD:` related to continuous integration/deployment
- `: data access:` related to data access functionality
- `: hub admin:` related to hub administrator functionality

2.7 How to update the team compass

The Team Compass should be updated frequently as information about 2i2c changes. Do not hesitate to propose a change! This page contains instructions for how to build the Team Compass locally, and how to update the website.

The Team Compass is built using Sphinx, along with themes and extensions that are used by the [Jupyter Book project](#). It is hosted via GitHub Pages and changes are auto-deployed via GitHub Actions.

2.7.1 Propose a change to the team compass

If you'd like to update information in this book, you can do so via editing the book's source files. These are a collection of markdown files in [the book's repository](#).

To do so, you have two options

Option 1: Use the GitHub UI

The easiest way to propose a change to the Team Compass is by using the GitHub User Interface. To do so, navigate to the GitHub page for a file that you'd like to edit, and click on the pencil () icon.

Make whatever changes you like using the GitHub interactive editor, and then click **Create a new branch for this commit and start a pull request..** This will create a Pull Request with your proposed changes.

Option 2: Make the change locally

The other option is to make the change locally on your computer, push it to GitHub, and then propose a Pull Request. To do this, you should be familiar with `git`. To do so, follow these steps:

1. **Clone the source files.** First, clone the source files for the Team Compass:

```
git clone https://github.com/2i2c-org/team-compass
cd team-compass
pip install -r requirements.txt
```

2. **Make your changes.** Edit the markdown files in the Team Compass and the content will be updated.
3. **(optional) Preview your changes.** To preview the book locally, build it with Sphinx:

```
make html
```

This will build the book using Sphinx, and put HTML outputs in `_build/html`. You can then preview them by opening up one of the `.html` files in a web browser.

MANAGED JUPYTERHUB SERVICE

The Managed JupyterHub Service is a special ongoing project of 2i2c. This section contains information about the service, as well as ongoing updates about where information is stored.

3.1 The Managed JupyterHub Service

The Managed JupyterHub Service is a special project that is run by 2i2c. It is an ongoing service, and thus is less development-oriented than projects that are funded by grants and collaborations.

The key goal of the 2i2c Managed JupyterHub Service is to launch a self-sustaining service that provides managed JupyterHub distributions to customers in research and education.

We are currently in the pilot phase of this project, with the goal of creating a basic shared infrastructure that can deploy, configure, and manage many JupyterHub distributions. We are also building a sales and support pipeline around this infrastructure.

3.1.1 Where is information located?

Infrastructure and development happens in [the pilot-hubs/ repository](#). This is both the deployment infrastructure for the 2i2c JupyterHubs, as well as documentation and team coordination around developing and running them.

User Documentation is located in [the pilot/ repository](#). This contains user-facing information about the Pilot Hubs, such as how they can add/remove users, update their environment, get support, etc.

3.1.2 A list of running JupyterHubs

We keep a table with all of our currently-running JupyterHubs at this location: [List of Running JupyterHubs](#).

3.1.3 Other project information

- [this google folder](#) has most information and brainstorms regarding this project
- [this github issue](#) also has several relevant links and discussion points

3.2 Pricing strategy

Currently, we are pricing 2i2c Managed Hub Services in a bespoke fashion, according to the needs and budget of each group that we work with. Over time, we plan on standardizing prices for various kinds of hubs (or at least standardizing the pricing process).

Our pricing approach and strategy is currently documented in the [Managed Service Plan](#).

3.3 Leads, sales, and hubs process

2i2c pursues new collaborations and contracts via following-up on leads for those who may be interested in 2i2c services. This page has information to help guide this process.

3.3.1 Location of sales assets and documents

Here are some locations of common documents used in the sales process (other than the repositories described below).

The Managed Hub Service folder Contains all folders and information that cover the Managed Hub Service, including sales and contracting.

The leads folder Contains a folder for each lead that 2i2c is currently engaging. This is where we store client-specific information, such as price quotes or contracts that have been sent to them.

The sales assets folder Contains all templates, email copy, stationery, etc that is useful in 2i2c sales.

3.3.2 The Leads to Sales workflow

Overview

We use Monday.com to track our leads **This Monday Board** keeps track of all leads in the sales process. Below is an overview of each step in the process.

Status fields indicate each step of the sales process We use Monday **status fields** to indicate where we are in the leads process. For example, a new lead is in the “Send Questionnaire” status of the “Triage Phase”. This means that the next action is to send them a questionnaire.

Fill in information following each status column As leads progress through the sales process, we need more information from them, eventually leading to a contract. Fill in this information, and once we reach the next “status” column on the board, we have entered a new phase of the process. At the end of a sale (AKA, once an invoice is sent and paid), all columns of the row should be filled.

Triage stage

The first stage is to understand whether an interested party is in-scope for 2i2c, and a good fit for the Managed Hub Service.

Create a lead When others show obvious interest in working with 2i2c, they are now a lead. **Create a new item in the sales pipeline** and fill in the relevant information for them (everything up to “triage status”).

Send the questionnaire To understand whether a lead is a good fit, we’ve created a [short questionnaire](#) for them. Send a link to the questionnaire.

Create a Lead Folder for them As soon as a lead responds to the questionnaire, create a folder for them in the [leads folder](#). Use this to store any lead-specific information, such as price quotes and contracts.

Decide if they are in-scope If the lead is in-scope, then set the Triage status column to “In Scope” and move on to the negotiation stage.

Negotiation stage

The negotiation stage is where we understand a bit more about what the lead needs and whether we’ll be able to provide it. For example, do they want something really custom that goes beyond our standard Managed Hub distributions? If so, they might be a “development lead” rather than a “managed service lead”.

Create a quote Create a lead-specific copy of the [pricing sheet template](#) as well as the [price quote template](#). Using the pricing sheet, and the answers from their questionnaire, create a price quote for them about the service.

Send a quote Once you’ve generated a quote for them, send them the modified Price Quote Template, along with a link to the [Managed Service Plan](#).

Warning: At this point, ICSI should be notified of the lead, because we have sent a specific price quote. This *should* be automated with [Monday.com](#).

Negotiate on a price We may require a few back-and-forths to clarify the needs of the lead and arrive at a specific number and date for service. Modify the Price Sheet as well as the Price Quote template as-needed throughout these conversations. If a customer agrees to a quote, we then send them a Contract.

Send them a contract Once a customer agrees to a price quote, send them a contract. Notify ICSI contracting as well as ICSI finance that we need to send this contract to the customer. They will prepare the contract in DocuSign and send it to them. Once they sign the contract, put a copy in the lead’s folder.

Note: This process may involve some negotiation between ICSI admin and the lead. Over time, we should detect common changes that clients want, and consider baking them into our contract language.

Send an invoice and get to work Once a customer has signed the contract, we may begin work on setting up their hub infrastructure, as well as sending them an invoice. ICSI finance will take care of the latter. We should next focus on setting up their hub.

The Hub Setup stage

The final step of the Leads and Sales process is to set up a managed JupyterHub for the lead (now, the “client”). To do so, follow the process below:

Create an issue for the client We use [issues in the pilot-hubs/ repository](#) with the *Hub* label to track the progress of each hub we are running.

The New Hub generation process Follow the [process defined in the pilot-hubs repo](#) in creating a new hub for a lead. This will then track the operation and maintenance of the hub.

To Do: Create a Customer Management system

In addition to tracking our sales / leads, we need to track running hubs and active customers in order to quickly understand their status and respond to their needs.

3.3.3 Tips and FAQs

When to decide to work with a lead

To decide whether we should continue engaging with a particular lead, we need to know whether they are in-scope for our mission and likely to be able to successfully collaborate with us. The **MANIAC-T framework** can be helpful in coming to a decision. Here is a short description of these guidelines:

- **Money.** Is there a specific amount of money set aside for the project? If they are seeking funding in order to pay for this, what are the details?
- **Authority.** Does the contact have the authority to make a buying decision? Who else will need to be involved in the decision-making process?
- **Need.** What are their clear pain-points with their current infrastructure setup? Do they seem to have a real need for change?
- **Impending Event.** Do they have an upcoming event that is urgent or pressing?
- **Application.** Is their use-case a good fit for 2i2c infrastructure? Would their use-case require major changes to our hubs? Is there a different organization better-suited to help them?
- **Competition.** What alternative options are they considering? e.g., other organizations, internal approaches, or deciding to do nothing?
- **Timeline.** What does the buying process look like for this prospect? E.g., finalizing a contract, or receiving a sub-award.

TEAM REFERENCE

Contains information about the 2i2c team and our projects, and some useful resources to learn more.

4.1 Code of Conduct

Our goal is to create one of the best communities in the world for learning, using, and creating open infrastructure for interactive computing. Our mission is to build this community in a way that promotes fairness and justice for all - especially those from traditionally marginalized groups. This allows every member of our community to thrive and to make the biggest positive impact in their work and on others. Our Code of Conduct defines expected behavior and guidelines that help create such a community.

Accordingly, anyone who participates in a 2i2c space is expected to show respect and courtesy to others in all interactions, whether in GitHub repositories, our Slack channel, in in-person events, when representing 2i2c in public, or in events and spaces associated with *ICSI*.

To make sure that everyone has a common understanding of “show respect and courtesy to each other,” we have adopted the following code of conduct. The code of conduct is enforced by the *2i2c Executive Director* and *the Steering Council*.

4.1.1 Unacceptable behavior

The following types of behavior are unacceptable in 2i2c spaces, both online and in-person, and constitute code of conduct violations.

Abusive behavior

Harassment: including offensive verbal comments related to gender, sexual orientation, disability, physical appearance, body size, race, or religion, as well as sexual images in public spaces, deliberate intimidation, stalking, following, harassing photography or recording, inappropriate physical contact, and unwelcome sexual or romantic attention.

Threats: threatening someone physically or verbally. For example, threatening to publicize sensitive information about someone’s personal life.

Unwelcoming behavior

Blatant -isms: saying things that assume negative characteristics in a blanket fashion because of identification with a particular group. This is especially true for -isms around traditionally marginalized groups (e.g., explicitly racist, sexist, homophobic, or transphobic statements) For example, arguing that some people are less intelligent because of their gender, race, religion. *Subtle -isms* and small mistakes made in conversation are not code of conduct violations. However, repeating something after it has been pointed out to you that you broke a social rule, or antagonizing or arguing with someone who has pointed out your subtle -ism is considered unwelcoming behavior, and is not allowed at 2i2c.

Maliciousness towards other community members: deliberately attempting to make others feel bad, name-calling, singling out others for derision or exclusion. For example, telling someone they're not a real programmer or that they don't belong at 2i2c. If somebody makes such a statement without malice, they may still be in violation of the Code of Conduct if their actions are deemed especially/repeatedly unpleasant (see below).

Being especially or repeatedly unpleasant: for example, if we've received reports from multiple 2i2c users, team members, or collaborators of agitating, rude, or especially distracting behavior over an extended period of time.

4.1.2 Scope

2i2c community members are held to the standards outlined in this code of conduct when interacting in the 2i2c Slack or GitHub repositories, when interacting in-person at events where they could represent 2i2c (this is most professional events), in physical spaces with other 2i2c team members or collaborators, or in any *ICSI* space.

In addition, the 2i2c community and experience often extends outside those spaces—2i2c community members may go on walks together to get lunch, attend meetups or conferences as a group, communicate on social media, or interact with each other in other communities. Abusive or unwelcoming behavior between community members still has a profound impact on individuals and on the community when it happens beyond our walls. The 2i2c Executive Director and the Steering Council will use our discretion when deciding whether to enforce this code of conduct and potentially remove someone from the 2i2c community after reports of such behavior happening outside of 2i2c, taking into account the impact on the individual community members involved as well as the impact on the community at large.

The 2i2c Code of Conduct does not apply to interactions between users of a Managed JupyterHub, though we encourage leaders in those communities to adopt a Code of Conduct for their hub infrastructure. The Code of Conduct does apply to any interaction between a user of a Managed JupyterHub and a 2i2c Team Member.

When in doubt, please [report unacceptable behavior](coc:reporting) to us. If someone's behavior outside of a 2i2c space makes you feel unsafe at 2i2c, that is absolutely relevant and actionable for us.

4.1.3 Enforcement

We've categorized unacceptable behavior into abuse and unwelcoming behavior in the section above.

If we witness or receive a report about abusive behavior, we will contact the perpetrator to have a conversation with them and verify what has transpired. We will *follow this response protocol*.

If we verify abusive behavior, they will be removed from the 2i2c community and, if applicable, their employment with 2i2c will be terminated. Their Slack account will be deactivated, and permissions will be removed from any 2i2c-related repositories. They will not be welcome in any physical or digital spaces covered by the 2i2c Code of Conduct.

If we verify unwelcome, but non-abusive behavior, we will have a conversation with the person so they understand the expectation that they not repeat the behavior or other behaviors that would violate the Code of Conduct a second time. See the follow-up protocol for more information.

This is the protocol that 2i2c Executive Director and Steering Council members will use to respond to reports of code of conduct violations.

CoC Report Response Protocol

When a report is submitted through [the submission form](#), an email alert will be sent to the Code of Conduct Stewards of 2i2c.

Within two business days of receiving an email alert, the Code of Conduct Stewards will:

Read the report to determine whether there has been a code of conduct violation

- If not, they will reply to the reporter, explain that our code of conduct was not violated, and suggest other remediation options (e.g., meeting to get advice on how to resolve an interpersonal conflict).
- If yes, the Director will determine whether they are the best person to respond to the situation, or if it is more appropriate to hand off the report to another Director or Steering Council member (for example, if someone else already has a relationship with the reporter or the accused).

Follow up with the reporter

- We'll email to acknowledge that we've received the report, and are taking action
- We'll ask any follow-up questions we need to better understand the situation
- We'll confirm that we can contact the accused individual

Follow up with the community member who violated our code of conduct

- We will reach out to the community member over email, let them know that we've received a report of a code of conduct violation, and invite them to speak to us about the incident in person if possible, over video chat if not.
- If the report was of abusive behavior, or the second report of unwelcoming behavior:
 - During our meeting we will tell the community member we're removing them from the community, and disable their accounts.
- If the report was of unwelcoming behavior:
 - We will explain how their behavior violates our code of conduct, and what we expect of them moving forward.
 - We will warn them that a second code of conduct violation will result in them being removed from the 2i2c community.

If the Director or Steering Council member someone finds a report of a code of conduct violation somewhere other than our reporting form (e.g. on Slack, in an anonymous note, or in-person), that person will fill out the reporting form with all the information they can, and then the same process described above will be followed.

If a report is made anonymously without an email address provided for follow-up, or if the reporter indicates that they do not give us permission to act on their report, we will unfortunately not be able to take any action.

4.1.4 Reporting

If you see a violation of our code of conduct, please [report it to the 2i2c Code of Conduct Stewards](#).

Why should I report?

- You are responsible for making 2i2c a safe and comfortable space for everyone. Everyone in our community shares this responsibility. 2i2c Team and Steering Council members are not around the online spaces or at 2i2c events all the time, so we cannot enforce the code of conduct without your help.
- The consequences for the 2i2c community of not reporting bad behavior outweigh the consequences for one person of reporting it. We sometimes hear “I don’t want X person to meet consequences because I told someone about their bad behavior.” Consider the impact on everyone else at 2i2c of letting their behavior continue unchecked.
- 2i2c only works as an open, participatory, community-driven community because of shared trust between community members. Reporting code of conduct violations helps us identify when this trust is broken, to prevent that from happening in the future.

Where and how to report

Please report all code of conduct violations using [our reporting form](#). This is a short Google Form that will be sent to any Steering Council member acting as a Code of Conduct Steward (see below for details). Alternatively, if you wish to report to one of the Code of Conduct stewards specifically, email them directly.

Who responds to violation reports

Any Steering Council members with the “Code of Conduct steward” role must monitor this reporting form. The minimal set of people that serve in this role are

- The Executive Director
- At least one other Steering Council member

The people serving in these roles will be listed on the 2i2c website’s Steering Council page. Anyone with this role must have training in violation response.

Confidentiality

We will keep all reports confidential, except if we’ve discussed with you and agreed otherwise. When we discuss incidents with people who are reported, we will anonymize details as much as we can to protect reporter privacy.

However, some incidents happen in one-on-one interactions, and even if the details are anonymized, the reported person may be able to guess who made the report. If you have concerns about retaliation or your personal safety, and do not want us to share the details of your report with anyone (including the perpetrator) please let us know explicitly in your report. Unfortunately, in that situation we won’t be able to conclude that an individual has violated the CoC based on this report alone..

In some cases we may decide to share an update about a major incident with 2i2c team members, or with the entire 2i2c community. If that’s the case, the identities of all victims and reporters will remain confidential unless those individuals instruct us otherwise.

4.1.5 Social rules

In addition to having a code of conduct, we have four lightweight *social rules*. The social rules are different and separate from the code of conduct. They help us create a better environment by giving names to counterproductive behavior and acting as a release valve so that frustration doesn't build up over time. We understand and anticipate people to unintentionally break the social rules from time to time. Doing this doesn't make you a bad person or a bad community member. When this happens, it's not a big deal. Just apologize and move on.

The enforcement provisions in this code of conduct do not apply to the social rules. We won't give you a strong warning or expel you from the 2i2c community just for breaking a social rule.

If you have any questions about any part of the code of conduct or social rules, please reach out to [any 2i2c team member](#).

Social Rules

2i2c has a few social rules. They help ensure that the 2i2c community lives up to [our values](#), which is a fundamental part of 2i2c's mission. They also create a friendly, intellectual environment where you can spend as much of your energy as possible on interactive computing, open infrastructure, learning, discovery, and collaboration.

The social rules are:

- No well-actually's
- No feigning surprise
- No backseat driving
- No subtle -isms

One thing that often confuses people about the social rules is that we expect people to break them from time to time. This means they're different and totally separate from [our code of conduct](#).

No well-actually's

Alice: I just installed Linux on my computer!

Bob: It's actually called GNU/Linux.

A well-actually is when you correct someone about something that's not relevant to the conversation or tangential to what they're trying to say. They're bad because they aren't helpful, break the flow of conversation, and focus attention on the person making the well actually.

This rule can be a bit tricky because there isn't a clear line between relevant to the conversation and not. Sometimes your correction might actually be necessary, and it could still come off as annoying when you make it. The best rule of thumb is, if you're not sure whether something needs to be said right now, hold off and see what happens. You can always say it later if it turns out there's no way for the conversation to move forward without your correction.

No feigning surprise

Dan: What's the command line?

Carol: Wait, you've never used the command line?

Feigned surprise is when you act surprised when someone doesn't know something. Responding with surprise in this situation makes people feel bad for not knowing things and less likely to ask questions in the future, which makes it harder for them to learn.

No feigning surprise isn't a great name. When someone acts surprised when you don't know something, it doesn't matter whether they're pretending to be surprised or actually surprised. The effect is the same: the next time you have a question, you're more likely to keep your mouth shut. An accurate name for this rule would be *no acting surprised when someone doesn't know something*, but it's a mouthful, and at this point, the current name has stuck.

No backseat driving

Bob: What's the best tool for data science with tables?

Alice: NumPy.

Eve: (*from across the room*) No, you should use Pandas. It's better.

Backseat driving is when you lob advice from across the room (or across the online chat) without really joining or engaging in a conversation. Because you haven't been participating in the conversation, it's easy to miss something important and give advice that's not actually helpful. Even if your advice is correct, it's rude to bust into a conversation without asking. If you overhear a conversation where you could be helpful, the best thing to do is to ask to join.

No subtle -isms

Carol: Windows is hard to use.

Bob: No way. Windows is so easy to use that even my mom can use it.

Subtle -isms are subtle expressions that assume negative characteristics in a blanket fashion because of identification with a particular group. This is especially true for -isms around traditionally marginalized groups (e.g., subtly racist, sexist, homophobic, or transphobic statements). They are not as blatant as *Blatant -isms*, and may not be intentional. They are small things that make others feel unwelcome, things that we all sometimes do by mistake. Subtle -isms make people feel like they don't belong in the 2i2c community.

Subtle -isms can also be things that you do instead of say. This includes things like boxing out the only woman at the whiteboard during a discussion or assuming someone isn't a programmer because of their race or gender.

The fourth social rule is more complicated than the others. Not everyone agrees on what constitutes a subtle -ism. Subtle -isms are baked into society in ways that can make them hard to recognize. And not everyone experiences subtle -isms in the same way: subtle homophobia won't hurt someone who's straight in the same way it hurts someone who's gay. Subtle -isms can also be intersectional (for instance, statements that imply negative characteristics to a particular combination of race and gender) in ways that can multiply the harm.

There's another part of *no subtle -isms*: If you see racism, sexism, etc. outside of 2i2c, please consider the welfare of marginalized groups in 2i2c spaces before bringing it in. It is important to make space for complex and difficult discussions relating to subtle -isms, but constant conversation about these topics - or publicly discussing particularly difficult or traumatic topics - can become exclusionary on its own. For example, people from oppressed groups often find discussions of racism, sexism, etc. particularly hard to tune out. Before bringing up these topics, consider their impact on other participants in 2i2c spaces, and consider whether to have them in "general" conversation spaces, or to create a private or "opt-in" space for this conversation. When in doubt, optimize for the welfare of marginalized groups in our spaces.

How do they work?

The social rules are lightweight. Breaking one doesn't make you a bad person. If someone says, "hey, you just feigned surprise," or "that's subtly sexist," don't worry. Just apologize, reflect for a second, and move on.

The social rules aren't for punishing people. They help make 2i2c a pleasant environment where everyone is free to be themselves, to tackle things outside their comfort zone, and to focus on creating, learning, and collaborating.

Code of conduct

The social rules don't cover harassment or discrimination. For that, we have a separate [code of conduct](#) enforced by the 2i2c Code of Conduct Stewards. All members of the 2i2c community are expected to abide by our code of conduct.

4.1.6 How we developed the code of conduct

We arrived at these policies by a combination of:

- Listening to feedback and suggestions we've heard from open source communities over many years
- Reading the codes of conduct of other organizations we find to be thoughtful (see some examples below)
- Considering our experiences in other communities and projects in the past

4.1.7 Other things that don't fit in to the code of conduct

When to seek help immediately

Instead of filling out a code of conduct violation report, please contact law enforcement directly to report criminal activity (e.g. physical assault, sexual assault, theft), or to report a dangerous physical situation (e.g. fire, serious injury, fear that someone will hurt themselves or someone else).

Getting help

If you or someone else at 2i2c is struggling and needs help, don't hesitate to reach out to the 2i2c Executive Director and [Steering Council members](#) in person or over email.

4.1.8 License

The 2i2c code of conduct is available under the terms of the [CC0 license](#).

Parts of it are based on the [Recurse Center Code of Conduct](#), the [Jupyter Code of Conduct](#), the [PyLadies Handbook](#), and the [example conference anti-harassment policy](#) on the Geek Feminism Wiki, created by the Ada Initiative and other volunteers. It also takes inspiration from [projectinclude's guidelines for Codes of Conduct](#).

The Recurse Center Code of Conduct and the Geek Feminism conference anti-harassment policy are available under the terms of the CC0 license. The Project Jupyter Code of Conduct and the PyLadies handbook is available under the terms of the [Creative Commons Attribution 3.0 Unported License](#).

4.2 Active Projects at 2i2c

Projects are focused work that 2i2c conducts over a long-ish period of time. They describe the major pieces work that drive our daily activities.

In general, projects should have:

1. A name / description
2. A location where deliverables and tasks are being tracked.

See also:

For more information about our workflow around projects and deliverables, see *How we coordinate and plan*.

Below is a list of active projects for 2i2c.

4.2.1 2i2c organizational launch

As 2i2c is quite young, it must first build an organizational foundation for itself. This is an ongoing effort to define structure, process, and governance of 2i2c so that it can grow and execute on its mission.

Tracking deliverables

There are a few places to track this project:

- [Issues in the 2i2c meta repository](#) often contain actionable items towards this project.
- The [organizational projects backlog](#) contains a list of deliverables we are currently working towards.

4.2.2 Launch of the Managed JupyterHub Service

The Managed JupyterHub Service is an ongoing special project, with the goal of sustaining itself and providing JupyterHub infrastructure to others in research and education.

Tracking deliverables

Since most of the [hubs we run](#) are managed out of the [2i2c-org/pilot-hubs](#) repository, deliverables are tracked in that repository as well.

4.2.3 Pangeo Hub Infrastructure development

The Pangeo project has considerable needs in both operating and developing hub infrastructure for their community. We are collaborating with them to provide both daily operations of their hubs, as well as help them in evolvin and improving their infrastructure to better-suit their scientific mission.

Tracking deliverables

This project is just beginning, and currently does not have a location to track its deliverables. However, you can find discussion of general Pangeo activity in [the Pangeo repository](#).

4.2.4 Jupyter Meets the Earth

Jupyter Meets the Earth is an NSF EarthCube project to develop Jupyter infrastructure in service of the earth sciences.

Tracking deliverables

The [pangeo-data/jupyter-earth](#) repository contains deliverables for this project. Look for the *JupyterHub* label.

4.2.5 Jupyter Book development

The Executable Books Project is an international collaboration to build open source tools that facilitate publishing computational narratives using the Jupyter ecosystem. Its activities are split across a variety of repositories in the [executablebooks/organization](#).

Tracking deliverables

- The [meta/](#) repository contains high-level conversation and strategy in the project. Most project planning and execution is carried out in the issues in [executablebooks/](#) repositories.
- The [jupyter-book](#) repository contains many issues that are the best place to get started finding things of interest to work on.
- The [executablebooks](#) feature voting board contains a table of issues users have “voted” for. It is a good guide for prioritizing work.

4.3 2i2c Team Calendar

Below is a calendar with some relevant events for the 2i2c Team and Community. You can find a [link to the team calendar here](#).

4.4 Inspiration for 2i2c’s practices

4.4.1 The Team Compass

This team-compass is inspired by the team compass repositories used across the Jupyter ecosystem. For example:

- [The JupyterHub team compass](#)
- [The JupyterLab team compass](#)

In addition, it is inspired by several open company handbooks from companies and organizations dedicated to transparency. In particular:

- [The GitLab Company handbook](#)
- [The Basecamp Company handbook](#)
- [The Valve company handbook](#)

4.4.2 Organizational influences

These are organizations that have influenced the structure and practices of 2i2c (both as organizations to emulate, as well as avoid emulating).

Organizational structure / culture

- [GitLab](#) - is a large open-core company that is innovative in its remote- and async-friendly organization as well as its transparency.
- [Basecamp](#) - is a remote-only company that has been successful for many years. They also publish much of their company culture/practices online.

Open core / SaaS inspiration

Interesting SaaS with an open-core offering.

- [Discourse](#) - offers an open source hosted community forum application. They do a good job of walking the line between community values and sustainability (e.g., see [the discourse about page](#))
- [RStudio](#) - creator of the popular RStudio IDE for the R programming language. RStudio is a B-corporation that does an excellent job of cultivating an open and inclusive community around its products and open source ecosystem.

For-profit companies with interesting histories that we should avoid becoming.

- [Wikia / Fandom](#) - similar to the Wikimedia Foundation but a for-profit company offering wiki hosting that is ad supported / paid.
- [Reddit](#) - used to be open source code / infrastructure before it was acquired by Conde Nast!

Academia-focused companies/non-profits

- [Ithaka](#) - is a non-profit organizations that provides scholarly services for the academic community.
- [Internet2](#) - provides a variety of infrastructure and technical services to the academic community. Their technical expertise has shifted over the years but they've done a great job of remaining relevant and useful partners.

4.4.3 Documentation

Here are a few excellent guides for writing good documentation:

- [The Divio documentation framework](#)
- [GitLab's documentation structure](#)

4.5 Important Terminology

Here are some helpful terms that we use at 2i2c.

Source of Truth

Single Source of Truth Distributed teams and open communities need to balance information across team members, and ensure that everyone is on the same page. For this reason, it is recommended to adopt a “single source of truth” for anything important. This is an authoritative source that everyone can look to in order to know the current status and plan for anything we do at 2i2c.

4.6 Archive of Engineering Team Sync Notes

In early iterations of 2i2c, we manually added team sync notes to this team compass. We now keep track of team syncs in GitHub Issues, but leave these notes here as a historical reference.

Check the pages below for team sync notes.

4.6.1 Team Sync 2021

2021-03-08

Chris Holdgraf

Thanks I'd like to give

- Thanks to Yuvi and Georgiana for giving feedback on the team process issue around goals!

Updates from last two weeks ✓

- I have mostly been working with ICSI to define the contract that we'll use
- I have worked with the 2i2c founding team to more explicitly define our governance and organizational structure!
<https://github.com/2i2c-org/team-compass/pull/38>
- Doing a bit of biz-dev trying to refine the [Managed Service Plan](#) that we're offering to people.

What I'm up to next

- Beyond finalizing those two issues, I'm going to work with Yuvi a bit to improve our documentation for the 2i2c Hubs!

Erik Sundell

Thanks I'd like to give

- Thanks Yuvi for your encouraging spirit (love bombing twitter)!
- Thanks Chris for your attentive leadership (clear and quick responses)!

Updates from last two weeks ✓

- I have taken some time off.
- I have done a bit more review work than usual and less development efforts of my own.
 - Z2JH feature `prePuller.hook.pullOnlyOnChanges` is now available though btw
- I have helped Ariel with an AWS deployment

What I'm up to next

- Z2JH security vulnerability in z2jh guide on how to setup a AWS EKS cluster to be corrected and communicated.
 - <https://github.com/orgs/jupyterhub/teams/jupyterhubteam/discussions/8>
- Changelog writeup for z2jh

2021-02-15

Chris Holdgraf

Thanks I'd like to give

- Cathryn and Jim have both been very helpful in helping me think through some high-level strategic questions about 2i2c!
- Ryan Abernathy and Georgiana both helped interview our job candidates, and their feedback was invaluable!

Updates from last two weeks ✓

- The last two weeks have all been about doing diligence on our applicants, and working with ICSI to understand how we can open up our first sales with them.

What I'm up to next

- The work with ICSI is ongoing, but I think that we are getting closer. I've got a draft of a contract that they're sending to a lawyer soon, and then we should be able to start accepting money for running hubs.
- In addition, I hope to hear back from the two job candidates in the next few weeks about their decision!

Erik Sundell

Thanks I'd like to give

- I'm thankful for all 2i2c members for their efforts into building this org! :heart:
- I'm thankful for Yuvi's positive and uplifting comments and help reviewing various PRs! :heart:

Updates from last two weeks ✓

- I've done a little bit of this and a little bit of that across the JupyterHub org (Example [general maintenance PR](#)), but I'm currently most excited about [this pr](#). It is finally getting the z2jh configuration reference updated to fully cover the available config options and at the same time including a jsonschema file that helm can use to catch various config errors for users of the chart and provide sensible messages while doing so.

What I'm up to next

- Pushing onwards to z2jh 1.0.0, after the schema validation logic I want to acquire an “official status” on [artifactory.io](#) where it is listed and make do a pass at the z2jh documentation so it is updated with lessons learned across time.

Links to items for discussion

- @yuvipanda I've love to hear your suggestions on pod resource requests or empirical data on pod resource consumption as you hint you could share! To me, that would be one of those nice 1.0.0 features we can ship with the documentation.

I'd be very happy to format and writeup the documentation and such based on a more raw knowledge input btw.

Related: <https://github.com/jupyterhub/zero-to-jupyterhub-k8s/pull/2034>

Georgiana Dolocan

Thanks I'd like to give

- To Chris for giving me the opportunity to participate in the interview process, it was a great learning opportunity
- To Yuvi for explaining every technical concept so well.
- To Erik for all the work in z2jh and making jupyter-server-proxy fully compatible with JupyterLab3

Updates from last two weeks ✓

- These past weeks, I've worked on setting a new hub for the Mills College, upgrading the hub version in pilot-hubs as part of the process. I also tried to make the hub health checks more robust and investigated the use JupyterLab3 with the pilot-hubs.

What I'm up to next

- TLJH needs some <3, so I'll try to get it a bit more up to date
- Investigate if our current Auth0 setup can support multiple hub authentication methods

2021-01-27

Erik Sundell

Thanks I'd like to give

- Min put in a lot of effort to help me review PRs and it made me very happy!
- Yuvi pinged me about technical inspiration (database class where each user has a dedicated database available). It is inspiring, I'm excited about the possibilities that open up in educational settings by removing various technical barriers to get started without getting stuck in hardware setup / software installation / licenses etc.

Updates from last two weeks ✓

- I've worked on z2jh PRs towards 1.0.0, where the following updates can be relevant to know about:
 - [hub.extraFiles](#) / [singleuser.extraFiles](#) This feature can help 2i2c deployments by offloading you off the logic to do define volumes/volumeMounts and helping YAML/JSON/TOML file to be mounted be configured from multiple helm configuration files (config.yaml / secret.yaml). It is also possible to have standalone files but then you must use `--set-file` during `helm upgrade` which is a bit messier.
 - [seed secrets](#) + [followup fix](#) This feature makes us no longer need to set or pass `proxy.secretToken`, `hub.cookieSecret`, or `auth.state.cryptoKey` - they will be automatically generated if not explicitly set.

What I'm up to next

- Pushing onwards towards z2jh 1.0.0
- Write a paper with Ariel Rokem for PEARC about neurohackademy

Links to items for discussion

- I'd love help to get the [extraFiles](#) PR towards merge.
 - Thank you Chris and Yuvi for your previous review!! Your previous review points have been addressed now!
- I'd also appreciate a of the [seed secrets](#) [followup fix](#).
- I'm generally curious about developing a solution to providing a authentication proxy for JupyterHub's services, communicating with JupyterHub as an identity provider and relying on JupyterHub group membership something to decide if the user should be granted network access to the service - such as `/services/docs` for example. I'm not sure yet how to go about it, but I think it would be one of those somewhat general purpose tools and those motivate me a lot to work on!

Yuvi

Thanks I'd like to give

- To Georgiana for:
 - Taking full responsibility for working with Aaron Culich on the Mills hub
 - Writing full fledged health checks for hubs, so we have more confidence that they actually work
 - Fixing bugs with the UToronto hub as they get reported
- To chris, georgiana and ryan for helping run interviews
- To chris for organizing everything so we don't all drown

Chris Holdgraf

Thanks I'd like to give

- Thanks to Yuvi for giving Georgiana so much guidance on the Toronto hub!

Updates from last two weeks ✓

- Dealing with 100% parenting + 100% working (fun!)
- Working on moving forward some more contracts for 2i2c (see <https://github.com/2i2c-org/leads/projects/1>)
- Refactoring some of our team process / communication / workflow
- Setting up hiring practices that we can use for the OSIE hire + future ones!

What I'm up to next

- Figuring out our budget situation for the next 2 years
- Finding a way to spend my Jupyter Book grant!
- Thinking more about prices + features + products strategy
- Hiring somebody!

2021-01-05

Erik

Thanks I'd like to give

- Thank you everyone for connecting and establishing collaborations between 2i2c and so many amazing groups of people! I'm excited about 2i2c helping them!
- Thanks Yuvi for addressing the challenge of standardizing [Grafana Dashboards](#) for [z2jh](#) deployments!
- I'm very thankful for @manics work, most recently for creating and working with me on the [jupyterhub/action-k3s-helm](#) GitHub action we now reuse for four or more JupyterHub projects.

Updates from last two weeks ✓

- I've worked generally motivated by making JupyterHub repositories maintenance more sustainable by reducing complexity, improving CI, improving docs, and pushing for a 1.0.0 release of z2jh.

What I'm up to next

- Z2JH 0.11.0 release.
- Z2JH 1.0.0 features relying on Helm 3 to reduce complexity such as needing to manage proxy.secretToken etc.

Links to items for discussion

- Merging [this Z2JH PR](#) is what remains for the 0.11.0 release. I have looked for help to review it but the JupyterHub team members are generally low on capacity. I'm not happy about either of the options I see: to wait, to self merge, or to repeat requests for help.

Any ideas on how to manage this general situation and this specific PR is greatly appreciated. In general I currently wish for the JupyterHub team to compromise a bit on review stringency during times when reviewer's availability are low. It would be nice to have some guidelines.

- Meta discussion: Did I use this team-sync somewhat as intended? Any suggestions for changes in any form?

Geo

Thanks I'd like to give

- Many thanks to Yuvi for being a great mentor and for all the time and effort put in getting me familiar with the 2i2c projects and technology.
- Thanks for Chris' great work in growing 2i2c and for the great onboarding process created.

Updates from last two weeks ✓

- These past weeks, I spent most of the time working on getting familiar with the 2i2c pilot-hubs, fixing a couple of issues and reviewing a few PRs as part of the process.

What I'm up to next

- Supporting Yuvi in getting the UToronto hub ready for the winter semester.
- Developing tests to check and ensure the pilot-hubs' health.

Chris

Thanks I'd like to give

- Thanks to Erik and Geo for joining the first team (a)sync :-)

Updates from last two weeks ✓

- We've just submitted two NSF sub-awards, one for a satellite mission at Texas Tech (well, this one was a pre-award), and another for a collaboration with UW and others around collaborative cloud science!
- This has also involved a lot of discussion with ICSI about how to manage these collaborations etc.
- I have eaten a lot of and drank a lot of with my French family.

What I'm up to next

- Wrapping up the next few collaboration proposals
- Starting the hiring process for the open OSIE position
- Updating our team compass docs with things we've learned after the last few proposals we've written
- More about hub pricing and "products" that we offer

Links to items for discussion

- Just a general one - check out the team compass! <https://2i2c.org/team-compass> and please don't hesitate to contribute anything to this that you think would be useful.

4.6.2 Team Sync 2020

2020-12-18

Chris Holdgraf

Thanks I'd like to give

- Yuvi has helped me a bunch in figuring out technical scope of work / needs for a few collaborator proposals. Thanks Yuvi!

Updates from last two weeks ✓

- Most of my time these last few weeks has been spent on organizational stuff (see the #org-updates channel in the Slack for some updates there)
- I've been attending the Jupyter Meets the Earth bi-weekly meetings, and will help advise on documentation and using Jupyter Book
- We've also been in touch with the education team at Berkeley to work out a system of collaboration for community colleges that are using the educational hubs. Going to meet early next year to get things started.

What I'm up to next

- Finishing up some proposals for collaborations with other geoscience use-cases in 2i2c
- Finding some opportunities to align development between the JMTE grant and 2i2c

ABOUT 2I2C

Information about the broader 2i2c organization, as well as its mission and structure.

5.1 Basic information about 2i2c

This page contains information about the 2i2c organization and community.

5.1.1 2i2c's mission, vision, and values

Your best place to learn about 2i2c's values is to check out [the About pages](#) on the 2i2c website. Those give an idea for the kind of organization that 2i2c strives to be, and the impact that it wishes to have on the world.

5.1.2 2i2c's structure and governance

Check out *Structure and governance of 2i2c* for more information about this.

5.1.3 Where is 2i2c located?

While 2i2c is housed under the [International Computer Science Institute](#), located in Berkeley, CA, 2i2c team members are distributed across the globe and 2i2c claims no formal physical center. All of its work is done online.

5.1.4 2i2c's current team

You can find more information about 2i2c's current team in the [About page of its website](#).

5.1.5 What is 2i2c's legal designation?

2i2c is a non-profit initiative that is housed by [The International Computer Science Institute \(ICSI\)](#), a 501(c)(3) non-profit organization.

5.1.6 What kind of things does 2i2c do?

2i2c does several kinds of things:

- **Manage Infrastructure as a Service** for research and education, using 100% open source infrastructure, and that respect a customer's right to replicate. Currently, this is done via *Managed JupyterHub Distributions*.
- **Develop and maintain** open infrastructure and open-source tools that underlie these services.
- **Provide technical collaboration** with researchers and educators on focused projects.
- **Provide guidance and strategic assistance** to organizations trying to navigate their decisions around cloud and open-source infrastructure.
- **Provide OS community and project support** for the major open source projects that align with our mission and values.

5.1.7 Why is 2i2c a non-profit?

If 2i2c is effectively creating a business unit, so a reasonable question to ask is: why not start a company instead? We thought a lot about this, and ultimately came down to a few reasons:

1. There's no reason that non-profits can't generate revenue, as long as they do so in accordance with their mission.
2. We want to design 2i2c beyond its initial founding team. This means adding organizational (and legal!) constraints that ensure it aligns with its founding values.
3. We believe a non-profit is best-positioned to advocate for open source ecosystems as a relatively neutral third party.
4. We believe that being a non-profit makes a strong statement about our commitment to our mission and values, and that this will make us a more attractive partner to research and education.

5.1.8 The virtuous cycle we aim to enable

We wish to create a virtuous cycle between research and education. It goes something like this:

Research and education organizations need help with dev-ops, deployments, customizations, and navigating the cloud landscape. They want to use and support open source tools, but need assistance in accomplishing this.

Open source communities need support from organizations that are using their tools. Most organizations use these tools without contributing back substantially. For these communities to survive, we need more organizations committed to contributing back.

This leads to the virtuous cycle:

2i2c wishes to **support research and education** by providing managed services and development that use open source tools. This will **generate resources**, that it then funnels back into **open source support** for both communities and tools that underlie the infrastructure it creates. This will lead to better tools, which makes this stack more attractive for research and education, thus increasing demand for managed services and development.

5.1.9 The open source communities we work with

These are the open source communities that we are particularly invested in. As a general rule, we want most of our infrastructure improvements to happen via upstream contributions in these communities. Moreover, we encourage our engineers to spend part of their time doing broader community work (reviews, triage, community support, leadership) in these communities.

Note: This will be an ever-changing list as this ecosystem evolves!

- JupyterHub / Binder
- JupyterLab
- Jupyter core
- Dask
- ExecutableBooks

In addition, we will likely interface heavily with projects in the broader PyData ecosystem (e.g., [xarray](#) or [holoviz](#)).

5.2 Who is on the 2i2c Team

The 2i2c team is defined on the [about/](#) page of our website.

5.2.1 Engineering team locations and times

For a quick glance at which timezone each team member is in, see the [this whena.re website](#), or the iframe below.

5.3 Structure and governance of 2i2c

2i2c is an organization dedicated to supporting open infrastructure in interactive computing for research and education, as well as the open source communities that underlie this infrastructure. Its host organization is *International Computer Science Institute (ICSI)*.

This page describes the major organizational structures of 2i2c, and how they relate to governance and operations.

A few notes on governance

- All 2i2c members act as individuals, and not on behalf of their affiliated institutions.
 - All decisions made by the 2i2c Steering Council or Teams must abide by the policies of its host organization, *International Computer Science Institute (ICSI)*.
-

5.3.1 Steering Council

The Steering Council defines the mission, vision, and values of 2i2c. It also sets the strategic direction and priorities for 2i2c. The Steering Council provides oversight to the Executive Director of 2i2c and the Operations Team. The Steering Council group ([steering-council@2i2c.org] (<mailto:steering-council@2i2c.org>)) is the only “official” way to communicate with others on the Steering Council.

The Steering Council makes decisions via consensus (specifically, it strives for [rough consensus](#)). It is restricted by the legal obligations and policies of ICSI, 2i2c’s host organization.

See the [Proposal process section](#) for information about how the Steering Council makes decisions and conducts discussion.

See also:

The current Steering Council is [listed on the 2i2c website](#).

Proposal discussion and voting process

The following sections describe our process for officially communicating and proposing changes to 2i2c policy with the Steering Council.

How are proposals encoded?

There are two major places where each proposal is encoded/tracked:

- A Google Doc for the proposal language (and discussion). Each proposal should [follow this proposal template](#).
- An Issue in a GitHub repository to track the “to-do” item of discussing the proposal. Cross-link the issue with the Google Doc.

How do we discover and discuss proposals?

We use the [Steering Council group](#) to describe and discuss all proposals.

In addition:

- The Executive Director of 2i2c should provide regular (e.g., weekly) updates to the Steering Council email about active proposals or proposals that are currently being discussed and/or voted on.
- We maintain a [shared Google Calendar](#) that contains voting dates as well as Steerco meeting dates.

When does voting happen, and how long do we discuss?

Proposals should be open for feedback and discussion for a minimum of 2 weeks. After 2 weeks, the proposal author may decide to call a vote (if necessary) or to accept and implement the proposal (if a vote is not required). Any Steering Council member may request that the discussion period be extended.

Votes happen on either the first or the third week of the month. When a proposal is made, add 2 weeks to the date, then find the next voting day. This is the assumed day of the vote (or implementation, if a vote is not needed) unless requested otherwise by a steering council member.

How does voting work?

Votes are encoded in a table at the top of the document.

For proposals that require a vote, we track the “stage” of the discussion at the top of the Google Doc. Once a proposal is in the “voting” stage (where people are recording their votes), the proposal should not substantively change. Steering Council members then put their votes in this table.

How are proposals implemented?

If a vote passes, or if a proposal does not require a vote and enough time has passed to move forward, we implement it via changes to the Team Compass. The actions that implement a policy should be linked to from the Google Doc.

How are proposals archived?

After a decision has been made about the proposal (regardless of whether a vote was needed), then take these steps:

- If relevant, add links to any pull requests that implement the proposal (e.g., in the Team Compass).
- Move the proposal to [the proposal archive](#).

Note: We do not yet have a policy about whether / how proposals should be made public. For now, the proposal archive is only available to 2i2c team members.

What kind of changes must follow this process?

Policy changes that require a steering council vote or notification must follow this process. See the [Proposal Guidelines](#) below.

Policies and guidelines for consulting the Steering Council

The Steering Council provides valuable perspective and insight for strategic decisions of major importance. As such, the Executive Director should leverage the Steering Council by asking for its counsel on a regular basis, particularly for strategic and organizational matters.

In addition to generally consulting the Steering Council when it is helpful, there are also some situations where the Executive Director *should* or *must* consult the Steering Council and/or get approval before moving forward. This section describes those scenarios.

Decisions that require a full steering council vote

The [Steering Council](#) defines the strategy and major direction of 2i2c. It must vote on major decisions that have strong implications for 2i2c’s strategy or financial well-being. See [the proposal process section](#) for information about how it votes.

Here are some major decision areas that require a full steering council vote.

- Unplanned financial decisions over \$20,000 (e.g., deciding to hire a contractor that has not been written into a grant)
- Decisions that have significant implication for 2i2c’s financial health.

- Major changes in strategic direction and roadmaps for business or technical development.
- Changes to governance.
- Hiring Director-level positions within the 2i2c org.
- Defining salaries for Director-level positions within 2i2c.
- Changes to the Code of Conduct.

For other kinds of decisions, the *Executive Director* is given authority to decide.

Decisions that require notification, but not a vote

These decisions are significant or public-facing, and would benefit from SC input, but do not require a vote to move forward. The Steering Council should be notified via the @steering-council handle, and discussion left open for a reasonable amount of time so that the SC has time to give input. Any SC member can request a vote on any of these topics if they believe it requires consensus.

Here are some decisions that would require notification, but not a vote.

- Significant public-facing changes to the 2i2c website.
- Significant Human Resources (hiring, firing ,etc) decisions that were already planned.
- Operational policy for 2i2c staff (unless they have major financial implications).
- Decisions about grants to pursue and submit.
- Operational decisions around the “Managed Hub Service” business, within the strategic plan defined by the Steering Council.

Decisions that do not require notification/consultation

These are decisions that should be visible to the steering council, but not strictly required that they are consulted.

- In general, decisions that are more about execution of pre-defined strategy/goals/plans, rather than *changing* strategy, goals, and plans.
- Any decision that doesn't fit in the above categories.

To Do

These are items that we should clarify in future conversations:

- Create a job description of the Executive Director position
 - Create an annual review process + subcommittee for the Executive Director
-

5.3.2 Executive Director

The Executive Director oversees the execution of the mission and strategy provided by the Steering Council. They are the primary interface to *ICSI's* administration, and coordinate activity in the 2i2c Teams. The Executive Director oversees each team, and makes tie-breaking decisions if they are at an impasse in decision-making. The Executive Director reports to the Steering Council.

See also:

The current Executive Director of 2i2c is listed on the [Our Team page of the website](#).

5.3.3 2i2c Teams

2i2c Teams carry out the mission of 2i2c with a specific focus or project. They are made up of staff funded through ICSI, employees at other organizations providing in-kind support, or volunteers contributing their time to 2i2c. Each team has a decision-making process as well as a scope.

You can find a list of teams below.

- [2i2c Open Engineering Team](#)

2i2c Open Engineering Team

Team focus: Build infrastructure for the 2i2c Hub service, contributions to open source contributions for projects that underlie this infrastructure, and development of a bootstrap business model around that service.

Governance: *Lazy consensus-seeking*, or via delegation to individuals or groups within that team. This does not currently follow a formal process, but is made in good-faith conversation across the 2i2c repositories and issues.

5.3.4 International Computer Science Institute (ICSI)

ICSI is the host organization of 2i2c. It provides administrative and strategic resources for 2i2c and leadership to support them in their mission. ICSI supports 2i2c by assisting in managing partnerships with 2i2c collaborators and contracts with 2i2c customers, connecting with the larger open research and education community, fundraising, grant and financial management, assuming legal and financial risk for 2i2c, and hiring and management of staff.

5.4 Strategy and goals

Our Strategic Plan is a combination of active and aspirational priorities that help us get closer to achieving our organizational mission. These are broken down into **strategic goals and objects for each**. The *major projects of 2i2c* should all feed into this strategy.

Currently, 2i2c's Strategic Plan consists of a 1-year "bootstrap strategy". Our primary goal is to build a foundation for 2i2c and clarify out strategy and objectives for the coming years. This plan was created by the Founding Team of 2i2c and is located [here](#). This page is a summary and synthesis of the ideas in that document.

The following objectives should be accomplished by **October of 2021**.

5.4.1 Build a foundation for 2i2c's operations

Rationale:

2i2c needs an organizational home, structure, and governing body in order to begin exploring how it can best accomplish its mission. It also needs resources that can fund people's time to begin work.

Objectives:

- 2i2c has an organizational home that can sign contracts, hold liability, and administer our operations.
- 2i2c has a governing body and rules for governance.
- 2i2c has organizational funding for core staff for at least a year.
- 2i2c has hired core staff for its first year of operations
- 2i2c has organizational structure to handle both its engineering and the business operations.
- 2i2c has a clearly-defined Mission, Vision, and Values.
- 2i2c has a strategic plan for the next 1-3 years.

5.4.2 Launch the Managed JupyterHubs pilot

Rationale:

One of 2i2c's goals is to serve the research and education community through Managed Services that utilize infrastructure for interactive computing. It will take some exploration and learning to understand how to best-design these services. As a start, we wish to launch a **Managed JupyterHubs** service, and will begin with a pilot to explore this model.

Objectives:

- 2i2c manages JupyterHubs for at least two institutions.
- 2i2c manages more lightweight, community-specific JupyterHubs for several smaller groups in research and education.
- 2i2c manages a "generic" JupyterHub that is not tied to any single institution or group.
- 2i2c has a beta-level business model for the first iteration of our Managed JupyterHub service.
- 2i2c has built relationships with cloud providers that facilitate our ability to serve these hubs to our users.

5.4.3 Launch major collaborations

Rationale:

2i2c wishes to conduct focused development in collaboration with others in research and education. We wish to engage in several major projects that will support infrastructure that aligns with our mission, and that also feeds into our Managed JupyterHub service.

Objectives:

- 2i2c has launched a project around serving educational hub infrastructure to under-served educational institutions.
- 2i2c has launched a project to provide infrastructure development and support for the earth sciences community.
- 2i2c has made significant contributions across several open source communities that are related to the infrastructure it serves.

5.5 Positions and Salaries

This document represents the positions and salary levels that 2i2c uses. They are inspired by the hiring/salary structure of several tech companies¹.

5.5.1 Terminology

Each position at 2i2c takes the following form:

```
<role-name> L<level> - S<step>
```

For example, **Open Source Infrastructure Engineer L2 - S3**.

Here is a quick breakdown of this terminology and structure:

Role Positions are specific areas of focus and skills in 2i2c. The expectations and focus-areas within that role are set by Levels and Steps.

Levels Levels are large and distinct jumps in terms of area knowledge, role complexity and overall scope. Higher levels generally mean less-oversight and more autonomy, higher expectations of quality, more work on design / strategy than implementation, etc.

Steps Four steps are housed within each level and are meant to mark smaller milestones of growth in terms of ownership and initiative, as well as experience in the current role. These are generally not part of public materials, but are

5.5.2 Positions at 2i2c

Below is a list of positions that 2i2c currently uses.

Open Source Infrastructure Engineer

This role focuses on infrastructure that supports interactive computing. It intersects job titles such as “dev-ops engineer”, “site reliability engineer”, “software engineer”, and “cloud engineer”. This person should have experience with cloud infrastructure. They generally both operate 2i2c Hubs, as well as develop technology for them. In general, a *minority* of time should be spend operating hubs - we want to use operations as an opportunity to drive development and minimize human-in-the-loop maintenance.

Director of engineering

The director of engineering sets the technical strategy for 2i2c, as well as leads major technical initiatives for 2i2c infrastructure. Development work should be focused on architectural design and major infrastructure changes. This role defines the technical direction of 2i2c’s development efforts, and interfaces with 2i2c engineers to guide development in this direction.

¹ In particular, [Buffer](#), [Stack Overflow](#), and [Basecamp](#)

Director of 2i2c

This role is the director of 2i2c. They oversee strategy, execution, and fundraising at 2i2c. They also ensure that 2i2c engineers are empowered to execute on their responsibilities, and thrive in their positions.

5.5.3 Salaries

Salaries at 2i2c are determined solely by job title as well as several steps within that job (corresponding to experience in a particular job). Currently, the salary policy of 2i2c is [listed on the careers page of our website](#). We additionally have [internal salary documents](#) for the specific positions and levels.

INDEX

S

Single Source of Truth, **33**

Source of Truth, **33**