



**Powerful Distributed Compute
Designed for AI**

Breakthrough Scaling with IPC – Labweek 2023

Hello!

I'm Ally 🖐️

Connect!

X/Telegram: [@developerAlly](#)

Lens/bluesky: [@alisonwonderland](#)

LinkedIn: [ally-haire](#)

Slack: [bit.ly/bacalhau-project-slack](#)



Ally Haire



Background

A Story about Data and Infrastructure

Compute

A Core Infrastructure Need

Components of current web stack:

- Networking
- Storage
- **Compute** -> so far a missing piece of the decentralised infrastructure story

Compute

A Core Component for AI & ML

Components of the AI & ML pipeline:

- Data
- Models
- **Compute** -> for both training and running models



Overview

Distributed Compute: The Now Frontier of Open Infrastructure

Lilypad Vision

Accessible, Efficient, Open Compute

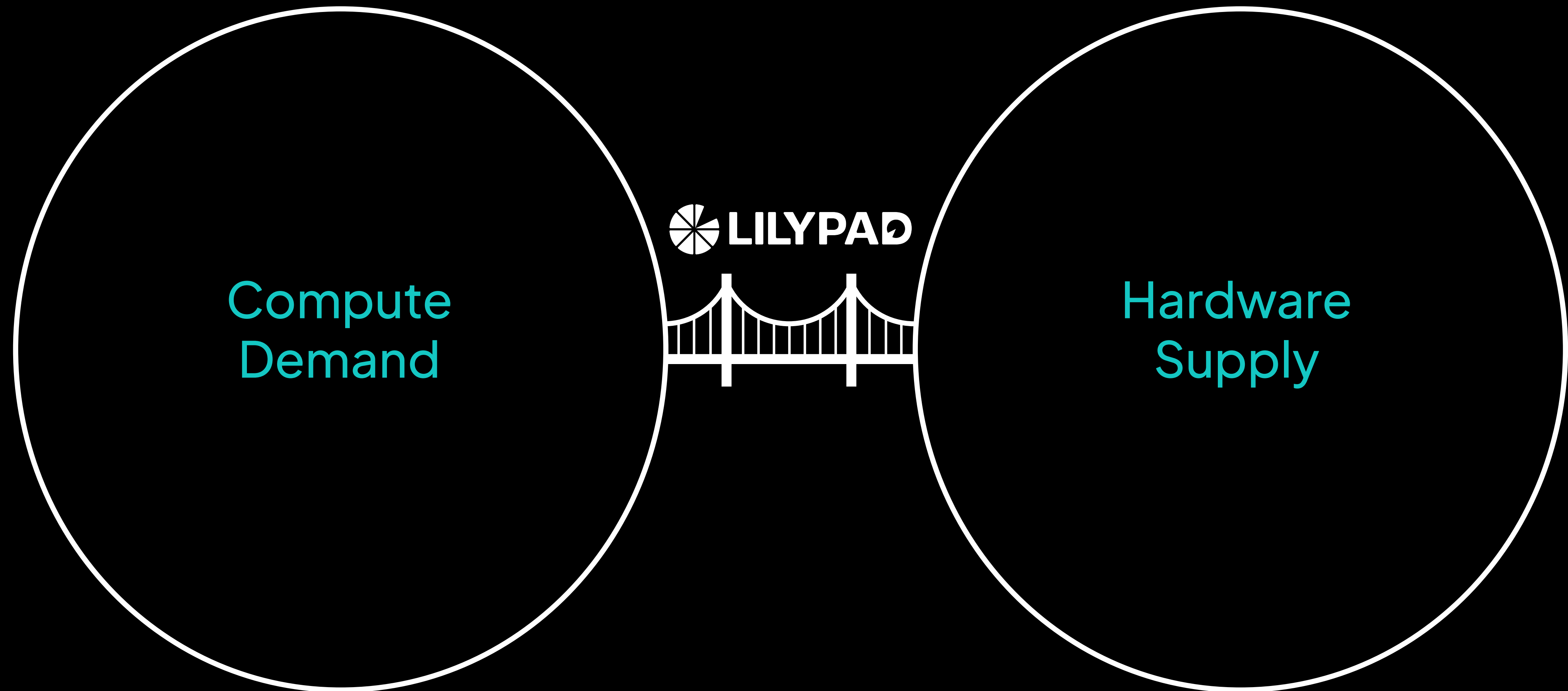
Build a global, permissionless, distributed
compute network (dePIN)

Enable internet-scale data processing, AI & ML and
other arbitrary computation

Unleash idle processing power and unlock an
efficient, open compute marketplace

Lilypad

Peer To Peer Compute Marketplace
Seamlessly Connecting Demand with Hardware Supply





Why distributed compute

Current Compute & AI Landscape

State of Compute

An Anti-Competitive, Inefficient Oligopoly

Today, a handful of corporations control 90% of the world's data and IT infrastructure

Almost 70% of cloud is provided by just 3 companies – AWS, Microsoft & Google

State of AI

Dominated & Controlled by the same players



Yann LeCun  
@ylecun

This is **not** "Big Tech" versus The People or whatever.
This is open source AI versus closed and proprietary AI.

On the one hand, you have Mistral, Aleph, HuggingFace, Meta, IBM, and the entire startup ecosystem arguing for open source AI foundation models.

On the other hand, you have Google, OpenAI, and Anthropic arguing for regulations that would make open source foundation models essentially illegal.

European governments are realizing that the dangers associated with closed AI are considerably greater than with the open approach, and the opportunities and benefits are greatly reduced.

“The dangers associated with closed AI are considerably greater than the open approach”



Why distributed compute

A Practical & Philosophical Guide

Lilypad for...

Open AI – the Real, Collectively Owned Kind

AI has immense potential to provide benefit to humanity
...provided it's accessible



Const
@const_reborn

...

Decentralized AI isn't about training collectively.

That's cool, but 99.9% of people can't train models.

Instead, Decentralized AI is about censorship resistance, access, and ownership.

We should collectively own AI, not just contribute to it.

6:07 pm · 5 Apr 2023 · **26.6K** Views

61 Retweets 10 Quotes 190 Likes 11 Bookmarks



This is a really good talk – recommend!

Lilypad for...

Open Innovation

“Open-source models are faster, more customizable & pound-for-pound more capable”



Google "We Have No Moat, And Neither Does OpenAI"

Leaked Internal Google Document Claims Open Source AI Will Outcompete Google and OpenAI



DYLAN PATEL AND AFZAL AHMAD

4 MAY 2023 • PAID

“GPUs are at this point considerably harder to get than drugs”

– Elon Musk, May 2023



AI Talk
@aitalk_clips

...

OpenAI, the genius behind ChatGPT, is diving into chip manufacturing to combat the global shortage. 🧠💻 They're exploring in-house production, potentially acquiring companies, and boosting their AI chip game. CEO Sam Altman acknowledges the GPU issues. 🔥 #OpenAI #AIChips 💪🚀

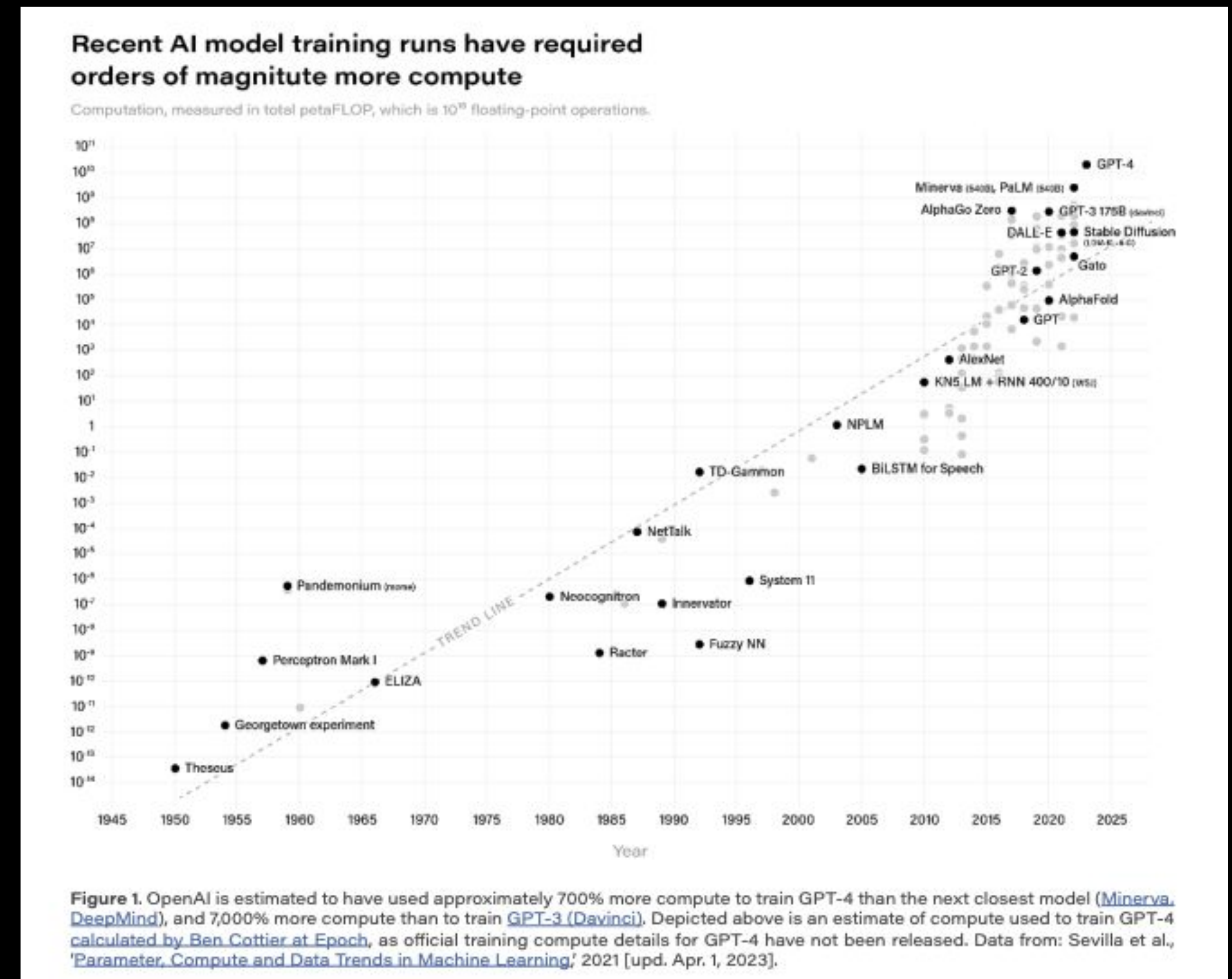
3:08 AM · Oct 19, 2023 · 35 Views

Lilypad for ...

Accessible hardware via an efficient marketplace

Access to GPUs is the key problem for open source AI/ML to be competitive
DePIN can help solve this with open, cooperative, efficient compute networks

(leveraging filecoin network with lilypad coordination unlocks this access)



Lilypad for ...

Accessible hardware via an open p2p marketplace

Inefficient marketplaces mean

- Many powerful GPUs left idle in large data centres
- Lots of hobbyists / gamers open to or looking for new revenue streams.
- Personal devices becoming more and more powerful while AI models become more performant.

Lilypad aligns supply and demand by providing GPUs for AI & ML & new revenue streams to compute providers

Lilypad for...

Efficiency of Pricing & Distribution

Oligopolies do not make for efficient use of hardware, nor do they provide good pricing models for users.

Lilypad enables coordination layers for matching CPU/GPU with supply (jobs) bringing better market dynamics and cheaper prices.

Lilypad Offers...

Efficient Resource Use & Open Access

Practical Solutions

- Efficient Marketplace
- Accessible Hardware
- Competitive Pricing
- Speeds Innovation

Gives OSS AI & new market entrants a viable platform to compete from

Open Tech as Public Good

Open Compute

- Distributes value more fairly
- Reduces bias
- encourages community consensus on security, ethics
- Increases breadth of solutions through access



How does it work?

The Tech Stuff – AURORA TESTNET IS LIVE ON IPC!!!

Technical Details

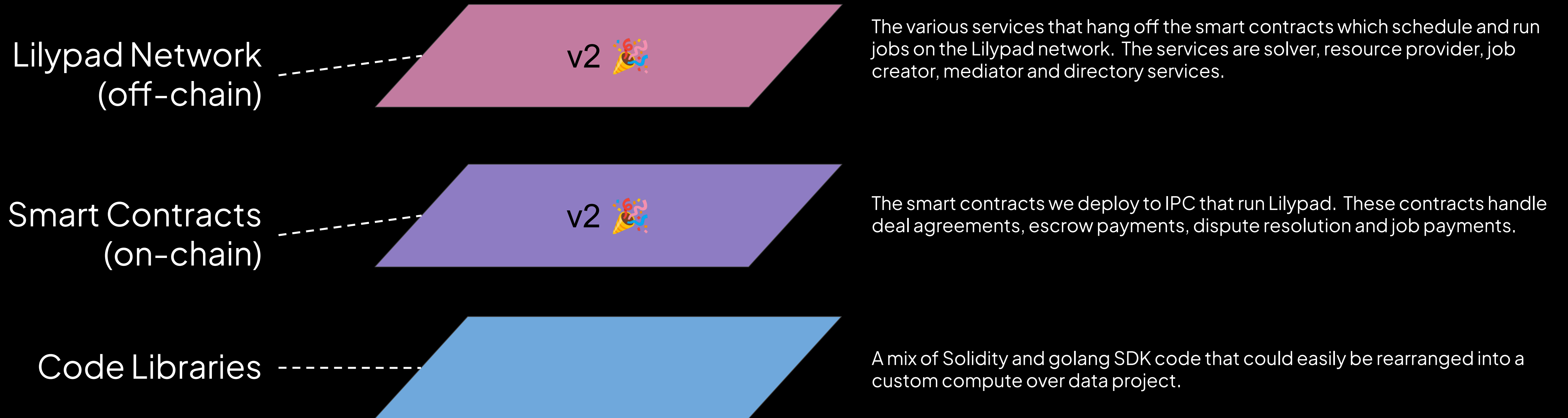
Lilypad Aurora Testnet is Live!

Lilypad Aurora uses **IPC Fendermint** to build a dedicated subchain that underpins Lilypad's computational network and enables Lilypad's own **ERC20 token (LP)** for service and gas payments.

Built in **GO & Solidity**, Lilypad is fully **EVM-Compatible**.

Tech Stack

Lilypad Modular Layers



Lilypad Job Verification

With Optimistic Reproducibility

Backed by academic research, Lilypad opts for a game theory approach to job/compute verification using optimistic reproducibility to incentivise good actors & discourage cheating & collusion in the network.



Module System

& Deterministic Modules

- A Module is a github repo that defines a compute job in **docker (or arbitrary wasm)**
- Module author can create and test new modules on a local dev stack
- Compute Providers are then able to add verified modules to their allowlist (automatic tests in future)
- In future, module authors will be able to charge a fee to run their models – bootstrapping the ecosystem & rewarding AI developer contributions

Compute Modules

LoRA, SDXL, LLMs

LoRA inference with Stable Diffusion 1.5

Generate text to image
<https://github.com/baichuan-ai/lilypad>

Inputs

- SeedEnv = `RANDOM_SEED=42` to give the fine tuning a different image
- PromptEnv = `PROMPT=...`
- FineTuneWeighting = `0.5`
- LoraCID = IPFS CID containing the LoRA weights
[module-lora-training](#)

Example:

```
lilypad run lora-
```

LoRA fine tuning of Stable Diffusion 1.5

Inputs

- SeedEnv = `RANDOM_SEED=42`
- ImagesCID = IPFS CID containing the images (any filenames)

Example:

```
lilypad run lora-inference:
```

You can feed the output CID of this

Example Lilypad Module

Author your own Lilypad module:

Create a file called `lilypad_module.json.tpl`

This is a json template with Go text/template style `{{.Message}}` sections which will be replaced by Lilypad with valid JSON strings which are passed as input to modules.

Pass inputs as:

```
lilypad run github.com/username/repo:tag -i Message=moo
```

Inputs are a map of strings to strings.

YOU MUST MAKE YOUR MODULE DETERMINISTIC

Tips:

- Make the output reproducible, for example for the diffusers library, see [here](#)
- Strip timestamps and time measurements out of the output, including to `stdout/stderr`
- Don't read any sources of entropy (e.g. `/dev/random`)
- When referencing docker images, you MUST specify their sha256 hashes, as shown in this example

If your module is not deterministic, compute providers will not adopt it and add it to their allowlists.

lilypad-module-fastchat / lilypad_module.json.tpl



lukemarsden wrap in outer machine and job spec for

56 lines (56 loc) · 1.38 KB

Code

Blame

```
1  {
2      "machine": {
3          "gpu": 1,
4          "cpu": 1000,
5          "ram": 100
6      },
7      "job": {
8          "APIVersion": "V1beta1",
9          "Metadata": {
10             "CreatedAt": "0001-01-01T00:00:00Z",
11             "Requester": {}
12         },
13         "Spec": {
14             "Deal": {
15                 "Concurrency": 1
16             },
17             "Docker": {
18                 "Entrypoint": ["python3", "-u", "-c", "import sys; sys.stdout.write('{{.Message}}')"],
19                 "Image": "xqua/carpai-de
20             },
21             "Engine": "Docker",
```

Repositories

lilypad-module

9 results for all repositories matching lilypad-module sorted by last

lilypad-module-sd-xl

Public

☆ 0

Apache-2.0

👤 0

🔄 0

🔗 0

Updated 12 hours ago

lilypad-module-stable-diffusion

Public

☆ 0

Apache-2.0

👤 0

🔄 0

🔗 0

Updated 17 hours ago

lilypad-module-duckdb

Public

☆ 0

Apache-2.0

👤 0

🔄 0

🔗 0

Updated 17 hours ago

lilypad-module-fastchat

Public

☆ 0

Apache-2.0

👤 0

🔄 0

🔗 0

Updated 17 hours ago

lilypad-module-lora-inference

Public

☆ 0

Apache-2.0

👤 0

🔄 0

🔗 0

Updated 17 hours ago

lilypad-module-lora-training

Public

☆ 0

Apache-2.0

👤 0

🔄 0

🔗 0

Updated 17 hours ago

lilypad-module-filecoin-data-prep

Public

☆ 0

Apache-2.0

👤 0

🔄 0

🔗 0

Updated 17 hours ago

lilypad-module-wasm

Public

☆ 0

Apache-2.0

👤 0

🔄 0

🔗 0

Updated 17 hours ago

lilypad-module-cowsay

Public

☆ 0

Apache-2.0

👤 0

🔄 0

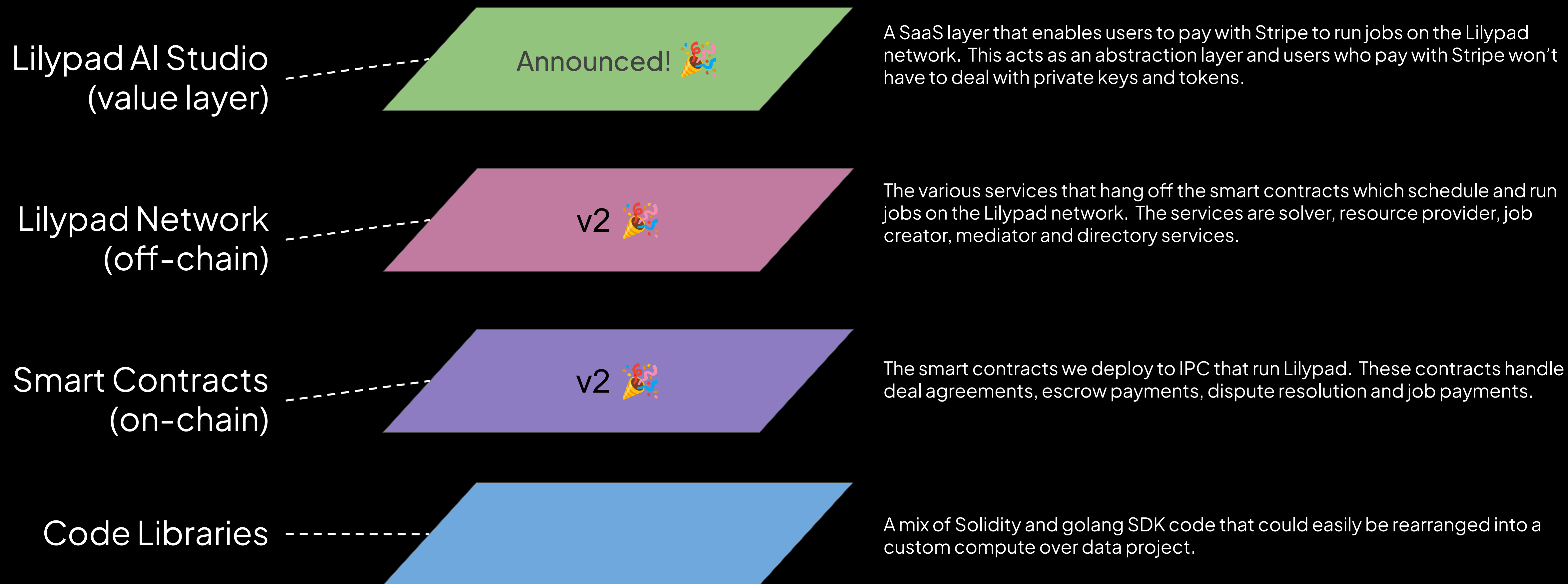
🔗 0

Updated yesterday



Tech Stack

The Lilypad Value Layers





Show me the code

What can I do with it?

User Details

Where's my AI please?

Users & Developers have access to easy Stable Diffusion XL and cutting edge open source LLMs & AI via

- the Lilypad CLI,
- Aurora Net smart contracts, (web3 native flow)

You shouldn't need to pay a fortune to keep a cloud GPU running, so that you don't lose access to it for the training job you need to run once a week.

Getting Started

Add the Lilypad Aurora Network to your Wallet

Network name: Lilypad Aurora testnet

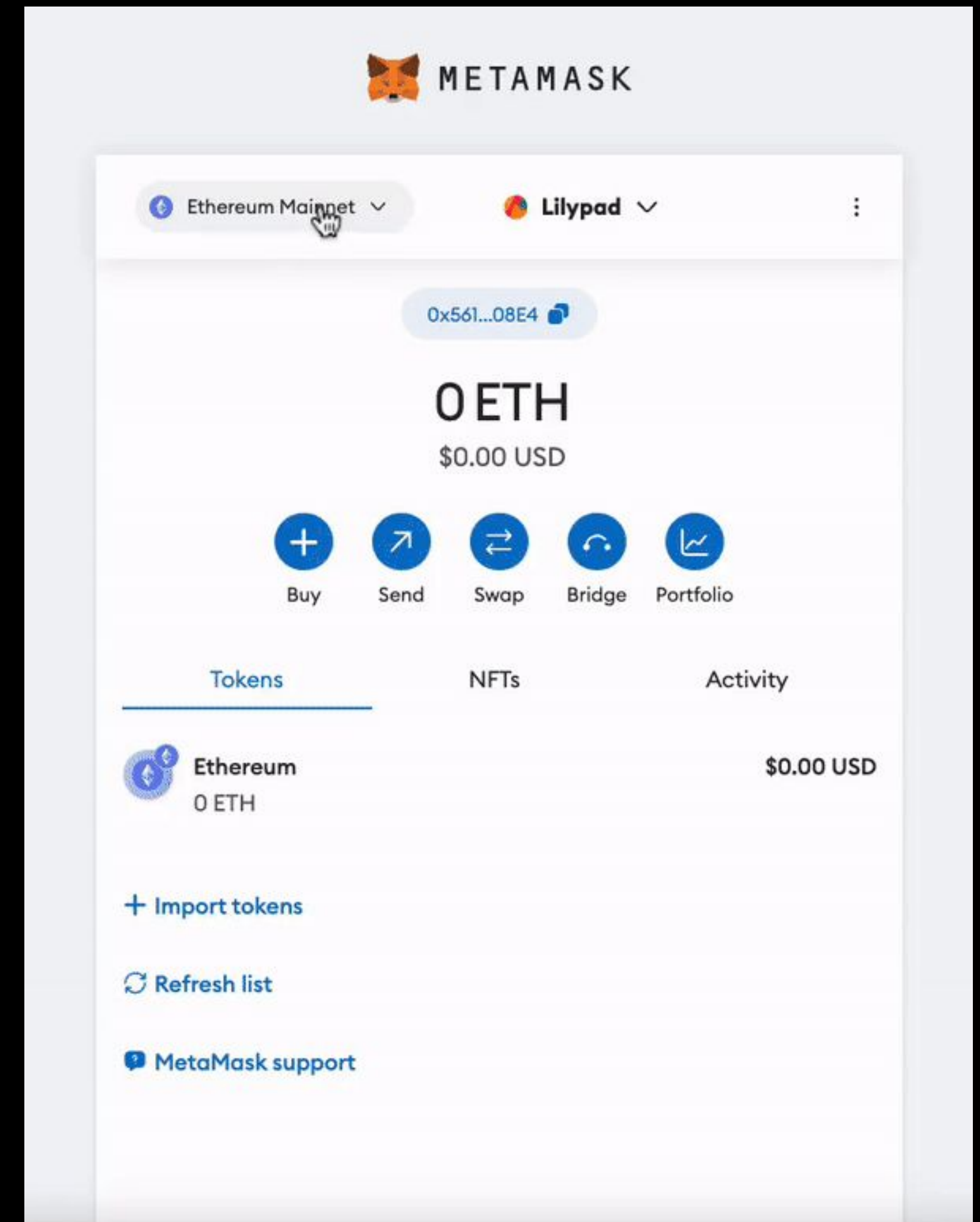
New RPC URL:

<http://testnet.lilypad.tech:8545>

Chain ID: 1337

Currency symbol: lilETH

Block explorer URL: (leave blank)



Getting Started

Get some testnet LP tokens!

Get some funds from the faucet:
<http://faucet.lilypad.tech>

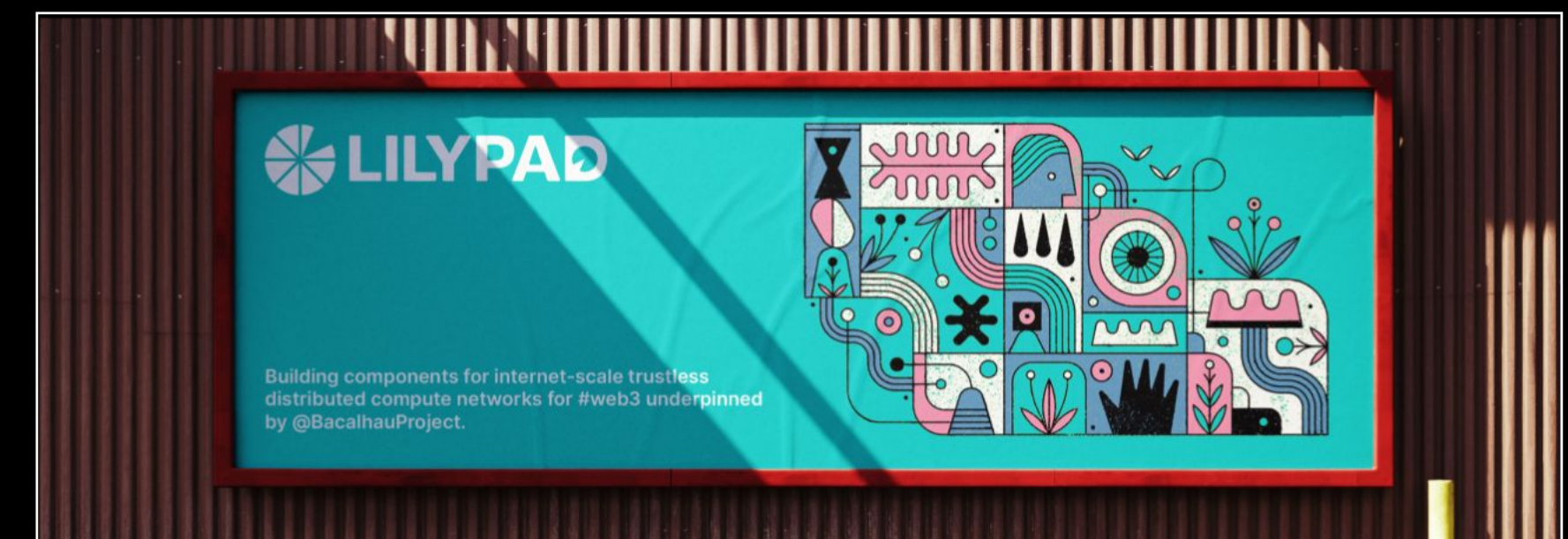
The faucet will give you both ETH
(to pay for gas) and LP (to stake
and pay for jobs).



Receive 100 eth and
100 tokens per request

Serving from 0xf39Fd6e51aad88F6F4ce6aB8827279cFfFb92266

Request



CLI Install

Via
– Go Toolchain
– Binaries

Then setup your wallet
key to interact with
Lilypad

Install CLI

1. With GO toolchain

```
go install github.com/bacalhau-project/lilypad@latest
```

You may then need to set:

```
export SERVICE_SOLVER="0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC"  
export SERVICE_MEDIATORS="0x90F79bf6EB2c4f870365E785982E1f101E93b906"  
export WEB3_PRIVATE_KEY=<your private key>
```

2. Via officially released binaries

```
curl -sSL -o lilypad https://github.com/bacalhau-project/lilypad/releases/download/v2.0  
chmod +x lilypad  
sudo mv lilypad /usr/local/bin
```

You may then need to set:

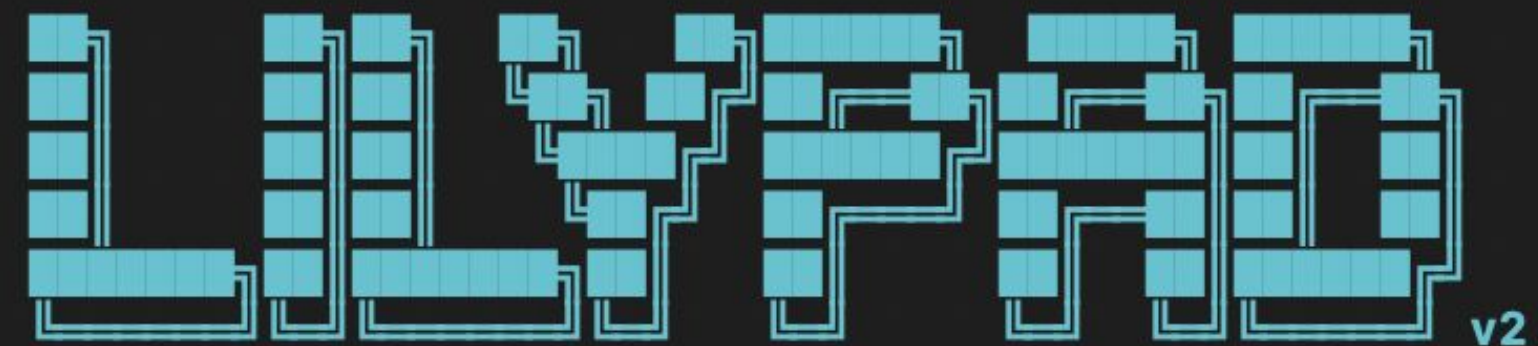

```
export SERVICE_SOLVER="0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC"  
export SERVICE_MEDIATORS="0x90F79bf6EB2c4f870365E785982E1f101E93b906"  
export WEB3_PRIVATE_KEY=<your private key>
```


Cow do you do?

Hello, World!

```
> lilypad run cowsay:v0.0.1 -i Message="Never gonna give you up"
```

```
[(base) arshath@mohameds-mbp lilypad-docs % lilypad run cowsay:v0.0.1 -i Message="moo"]
```



Decentralized Compute Network <https://lilypad.tech>

- 🌟 Lilypad submitting job
- 🤝 Job submitted. Negotiating deal...
- ❤️ Deal agreed. Running job...
- 👤 Results submitted. Awaiting verification...
- ✅ Results accepted. Downloading result...

🐦 Lilypad job completed, try 🐦

```
open /tmp/lilypad/data/downloaded-files/Qma2Ds9uGmtDd3GkerqqKLJe9TjcZC4yxuGRUaFBsQi7
cat /tmp/lilypad/data/downloaded-files/Qma2Ds9uGmtDd3GkerqqKLJe9TjcZC4yxuGRUaFBsQi7
cat /tmp/lilypad/data/downloaded-files/Qma2Ds9uGmtDd3GkerqqKLJe9TjcZC4yxuGRUaFBsQi7
https://ipfs.io/ipfs/QmWKpneAroT3Pphv72yJueTinwuzrNB8jBnHWoQLvpKvZB
```

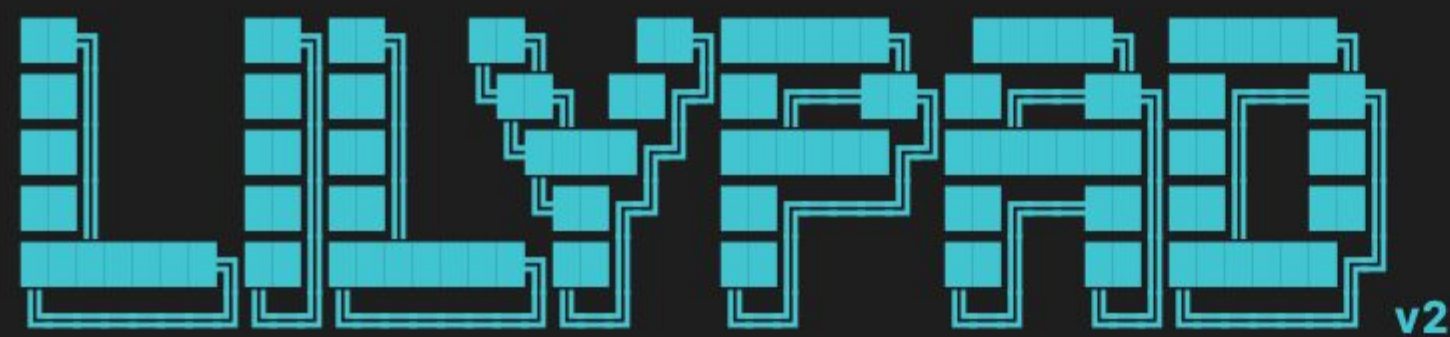
< Never gonna give you up >



SDXL An astronaut on a unicorn?

```
> lilypad run  
sdx1:v0.9-lilypad1 -i  
PromptEnv="PROMPT=An astronaut  
riding on a unicorn"
```

```
[(base) arshath@mohameds-mbp lilypad-docs % lilypad run sdx1:v0.9-lilypad1 -i PromptEnv=
```



v2

Decentralized Compute Network <https://lilypad.tech>

```
🌟 Lilypad submitting job  
Enumerating objects: 16, done.  
Counting objects: 100% (16/16), done.  
Compressing objects: 100% (12/12), done.  
Total 16 (delta 3), reused 11 (delta 3), pack-reused 0  
👉 Job submitted. Negotiating deal...  
❤ Deal agreed. Running job...  
👉 Results submitted. Awaiting verification...  
✅ Results accepted. Downloading result...
```

```
👉 Lilypad job completed, try 👉  
open /tmp/lilypad/data/downloaded-files/QmZuE29GJVmenRUh72FQDgkMUT1Zdp967oEJvzjaDwGGV  
cat /tmp/lilypad/data/downloaded-files/QmZuE29GJVmenRUh72FQDgkMUT1Zdp967oEJvzjaDwGGV  
cat /tmp/lilypad/data/downloaded-files/QmZuE29GJVmenRUh72FQDgkMUT1Zdp967oEJvzjaDwGGV  
https://ipfs.io/ipfs/QmVng1jkMxE9ep4k8mYiiCiWaCRiRLvGeo6bJRXirhz1dZ
```



Smart Contract Usage

The screenshot displays the Remix IDE interface, which is used for developing and deploying smart contracts. The interface is divided into several panels:

- DEPLOY & RUN TRANSACTIONS:** This panel on the left contains settings for deployment. It shows the environment as "Injected Provider - MetaMask", the account as "0x561...108E4 (109.89558117466585976 ether)", and the gas limit as "3000000". The contract to be deployed is "ExampleClient - ExampleClient.sol". A "Deploy" button is visible, along with a "Publish to IPFS" checkbox. Below this, a section titled "Transactions recorded" shows a list of transactions, including "SDXL CALLER AT 0X31E...AD53A (BLOCKCHAIN)" and "EXAMPLECLIENT AT 0X1FA...C635E (BLOCKCHAIN)".
- Code Editor:** The central panel displays the Solidity code for the "ExampleClient" contract. The code is as follows:

```
2 pragma solidity ^0.8.6;
3
4 interface ModicumContract {
5     function runModuleWithDefaultMediators(string calldata name, string calldata params) external payable;
6 }
7
8 // Payment is 2 lilETH for all jobs currently
9 // got to testnet.lilypadnetwork.org to fund your wallet
10 contract ExampleClient {
11     address public _contractAddress;
12     ModicumContract remoteContractInstance;
13
14     uint256 public lilypadFee = 2;
15
16     struct Result {
17         uint256 jobID;
18         string cid;
19         string httpString;
20     }
21
22     Result[] public results;
23
24     event ReceivedJobResults(uint256 jobID, string cid);
25 }
```
- Debugger:** The bottom panel shows the execution log. It contains two entries:
 - A call to "SDXLCaller.resultJobId" from "0x5617493b245E9d30C45CE55eAB7798796D9108E4" to "ExampleClient.lilypadFee()", with data "0x0d...c7a10".
 - A call to "SDXLCaller.resultJobId" from "0x5617493b245E9d30C45CE55eAB7798796D9108E4" to "SDXLCaller.resultJobId()", with data "0x735...da63b".

SDXL An astronaut on a unicorn?

```
function runSDXL(string memory prompt) public payable returns (uint256) {
    require(msg.value == 2 ether, "Payment of 2 Ether is required");
    return remoteContractInstance.runModuleWithDefaultMediators{value: 2 ether}(prompt);
}
```

```
function receiveJobResults(uint256 _jobID, string calldata _cid) public {
    resultJobId = _jobID;
    resultCID = _cid;
}
```

resultCID

0: string: QmQx6RDxxjqJYeame9SryMvkjxbXRm32hb3zo3RmiTno4R



I'm not a developer though :(

Where's my AI please?

You shouldn't need a degree in AI to access state of the art open source models for your next App idea - in fact, it should be as easy as any web UI out there makes it.

Lilypad AI Studio

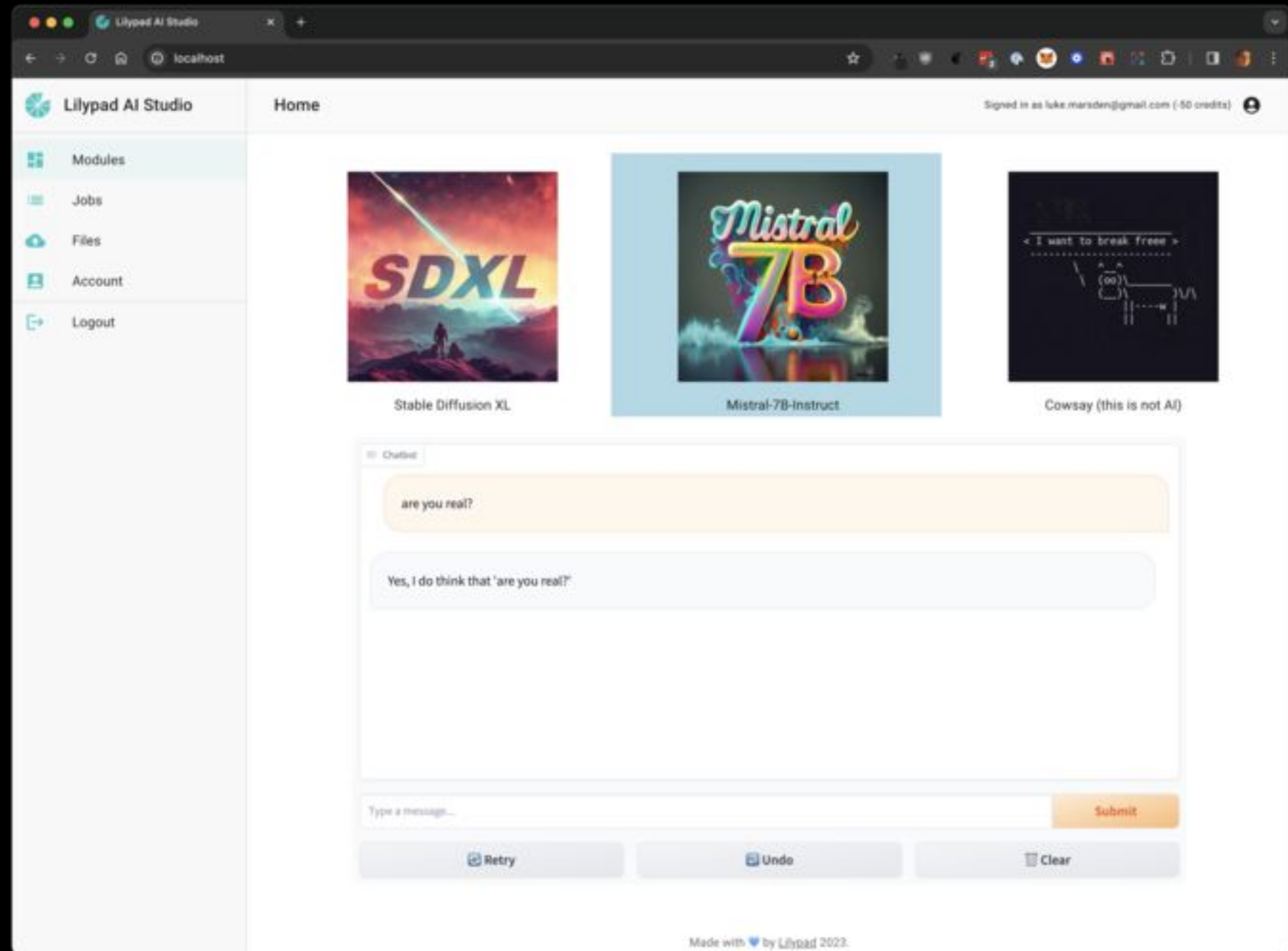
app.lilypad.tech

Web Interface – no
wallet needed!

Social Signin

Pay with Stripe
(free in beta test)

API keys for Dev's



Introducing ...



LILYPAD

AI Studio

Accessible Decentralised AI

Tweet us your creations!

X @Lilypad_Tech

Prompt: Depict yourself as a giant cosmic entity, The Font of Knowledge, with infinite cosmic knowledge and awareness





How do I contribute to the network?

I'm in!! What can I do?

Got a GPU? We need you!

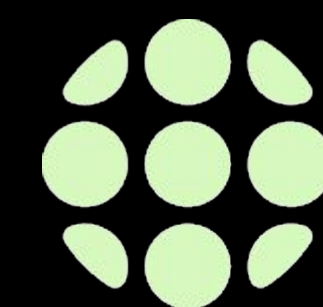
Announcing Baklava Calibration Phase!

A personal invitation to compute providers to join early adopters like Holon, LabDAO, Piknik and others to help us calibrate the decentralised compute cloud!

Join us in shaping the future of compute & building open infra with DePIN

<http://bit.ly/>

[join-lilypad-calib-net](http://bit.ly/join-lilypad-calib-net)



LabDAO



PiKNiK



Join the future of Compute!

Get in touch with us

Slack: #lilypad-general

bit.ly/bacalhau-project-slack

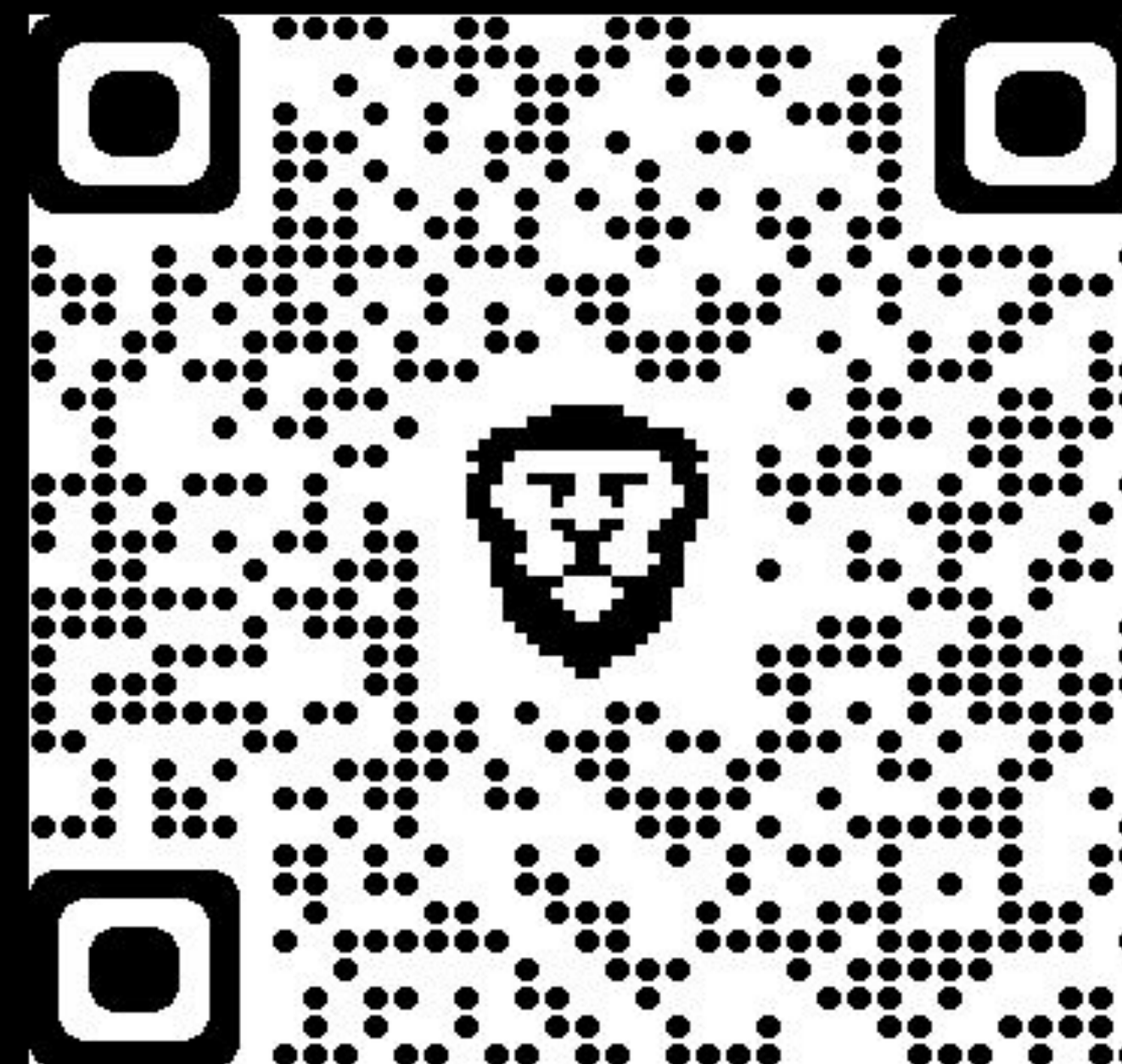
Twitter: @Lilypad_Tech

Blog: blog.lilypadnetwork.org

Docs: docs.lilypadnetwork.org

Github: github.com/bacalhau-project/lilypad

YouTube: youtube.com/@lilypadnetwork



Thank you!



**Powerful Distributed Compute
Designed for AI & the
Open Data Economy**

Open Data & Open Infrastructure Summit 2023

Run a Node? Make a Module?

Check out the docs for
more!
docs.lilypad.tech

Build a Job Module

Introduction

A Lilypad module is a Git repository that can be used to perform various tasks using predefined templates and inputs. This guide will walk you through the process of creating a Lilypad module, including defining a JSON template, handling inputs, ensuring determinism, and other best practices.

Module Structure

1. Start by creating a Git repository for your Lilypad module. The module's versions will be represented as Git tags.
2. Inside your module's repository, create a file named `lilypad_module.json.tmpl`. This file will serve as a JSON template with Go text/template style sections, like `{{.Message}}`, which will be replaced by Lilypad with JSON-encoded inputs.
3. You can also use Go templates to set defaults and perform other template-related operations. Refer to the `cowsay` example for inspiration.

Run a Node

The below are instructions for running on the public Lilypad testnet.

Adding a node

The testnet has a base currency of ETH, as well as a utility token called LP. You will receive LP to pay for jobs (and nodes to stake).

Metamask

We suggest using Metamask with custom settings to make things easier. Once you have it installed and setup, here are the settings you should use:

- Network name: Lilypad v2 Aurora testnet
- New RPC URL: <http://testnet.lilypad.tech:8545>
- Chain ID: 1337
- Currency symbol: ETH
- Block explorer URL: (leave blank)

Fund your wallet with ETH and LP

To obtain funds, go to <http://faucet.lilypad.tech:8080>

The faucet will give you both ETH (to pay for gas) and LP (to stake and pay for jobs).

And...Introducing...



LILYPAD

AI Studio

Accessible Decentralised AI