neptune.ai

We hire **great** Software Engineers

to develop an industry-standard system for ML training monitoring

The **ultimate guide** for **candidates** to **neptune.ai's Engineering Team**

About neptune.ai

Neptune is the most scalable experiment tracker for teams that train foundation models.

It consists of a Python API for logging billions of data points and a web app for visualizing them in seconds.

With Neptune, AI Researchers can monitor training and evaluate models faster, with less cognitive effort, while optimizing infrastructure usage.
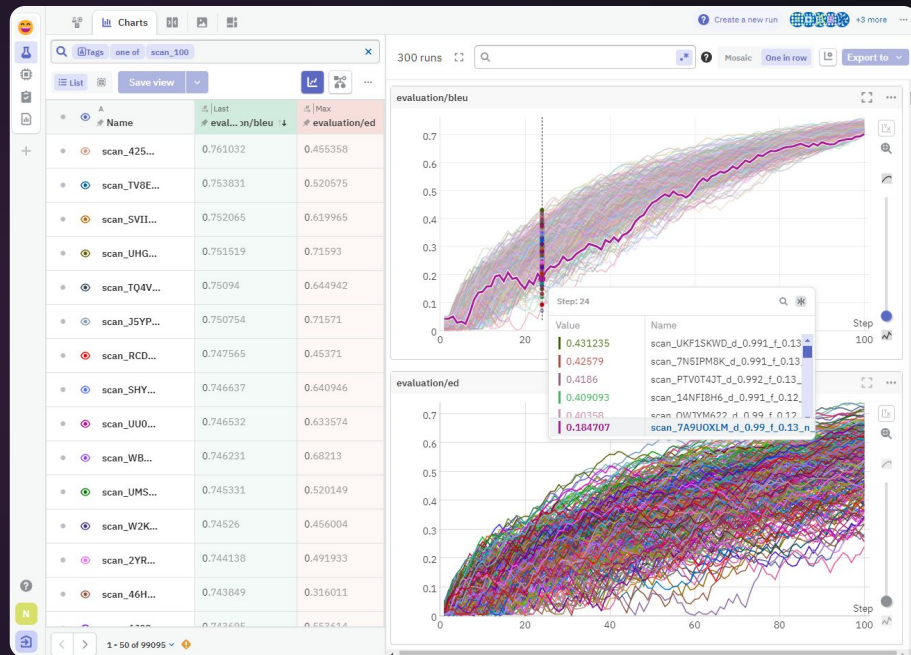
Unlike other experiment tracking solutions:

→ Neptune can ingest over 100k data points per second, enabling tracking of all metrics and metadata in one place;

→ Neptune's UI is quick and responsive at any scale, it displays metadata in milliseconds with 100% accuracy;

→ Neptune allows for forking experiments and supports the visualization of such experiments out of the box;

→ Neptune is available in a secure, scalable self-hosted version ready to be deployed from day one.

Backend

+

Web user interface

+

Programming Language library

+

30+ integrations with ML frameworks

neptune.ai is **a complex ecosystem** with many parts written in different technologies – developed by our **team of talented engineers**.
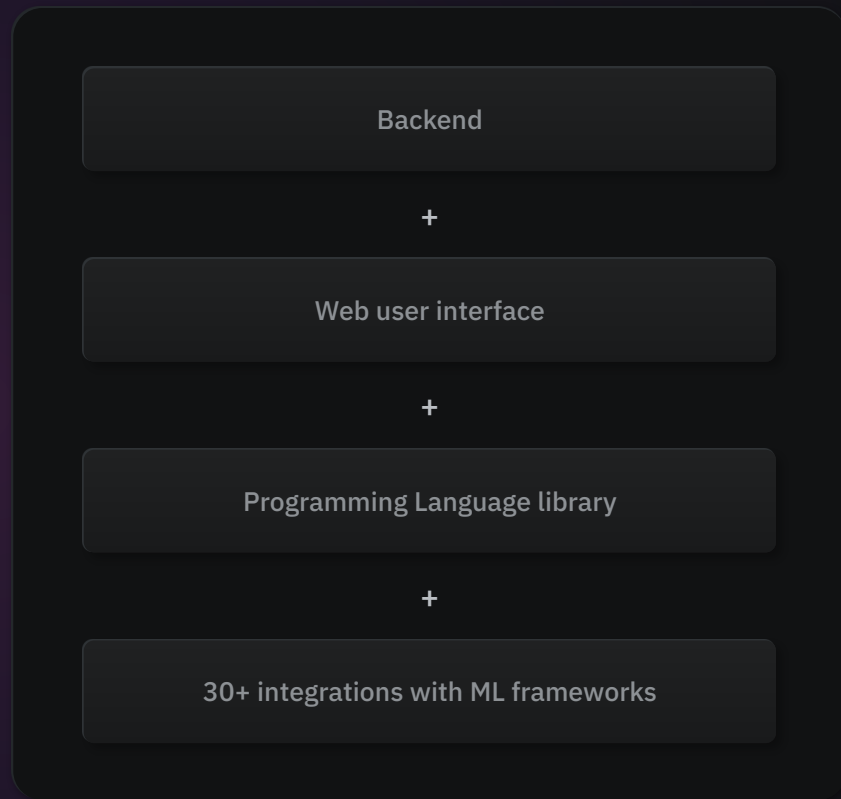
The story of Neptune starts in 2016 when we won Kaggle's Right Whale Recognition competition as a part of the deepsense.ai team.

We **spun-off in 2018** to focus on creating a system that manages the experimentation process – based on our experience.

Today, we are trusted by customers worldwide. The popularity and adoption of Neptune are growing quickly.

We are also trusted by investors – confirmed by investment rounds, with the latest of 8 million dollars in Series A.

Our investors:

AlmazCapital

b
t
o
v

TDJ PITANGO
ventures

RHEINGAU
FOUNDERS

70,000+
AI Researchers trusted Neptune

1,500+
Commercial
and research teams

30,000+
Projects tracked

BRAINLY

Roche

InstaDeep™

CAPE

BIOPTIMUS

poolside

SAMSUNG

workable

hp

GINKGO BIOWORKS
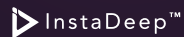
and more...

Short clip
intro to neptune.ai

youtube.com/2-min-demo

Product
live example

neptune.ai/example-project

People behind neptune.ai

We are a growing team of
50+ people, representing a few
nationalities spread across multiple
countries – Poland mainly, but also
Armenia, Egypt, France, Portugal,
Spain, Switzerland, UK, and more.

Check out our About us page to
e-meet us all!

Neptune is built by a **range of specialists**, including Software Engineers, Product Managers, Product Designers, Marketing Managers, Customer-facing Specialists, People and Operations Partners, and more!

As an organization, we're divided into **five teams**:

- Product
- Engineering
- Marketing
- Sales&Customer Success
- People&Operations

How we work

We are a **remote-first company**. The majority of Neptune's team is in the EMEA time zone, but there are exceptions, and most people are on a **flexible schedule**.

We meet in person, within teams, or as a whole company, two times a year – mainly to bond and have fun together.

We also have an **office space in Warsaw** (at Przeskok 2 Street) where we work together occasionally and coworking spaces in Wrocław, Poznań, and Kraków that you can visit whenever you want.

Communication is crucial for the company's success.

Although most of the staff is Polish, we keep all internal communication in **English**. Given our distributed nature, we support an **async type of communication**, with an emphasis on all meetings **bringing value**.

Once a month, we hold a **remote all-hands meeting** to share updates and news between all of the departments.

We use **industry-standard tools** to work effectively, including:

- Google Workspace,
- Slack,
- Notion,
- Linear,
- GitHub,
- GitLab,
- and other department-specific tools.

At neptune.ai, our approach to work is reflected in the **company's values**:

Own it

When you own something, it is on you, no excuses. We trust that if there is a need, you will learn, look at the data, get feedback, ask for help, and collaborate to make it happen. You get the freedom to do it your way.

Be concise, be intentional

Every word counts. We treat the reader's time as more valuable than our own; if it's not adding value, it isn't worth saying. Respect clarity, drive results, and leave no room for misunderstanding.

Make impact

Balance strategy and execution, learning and action to make the most impact. Understand why you are doing things but don't overthink it. Take action, but not without the skills that give you a chance. Remember, it is the impact that matters.

Iterate

We do the smallest thing possible and get it out as quickly as possible. Progress over perfection.

Engineering Team

The Engineering Team consists of **squads** working on separate parts of the system. Squads obviously cooperate – putting effort into maintaining **clear interfaces between components**.

Neptune is growing, and so is the Engineering Team. The team constantly evolves – over time, you can expect more **effective small-size subteams**.

Core Systems:

- Developing our backend services.
- Making sure our new APIs are well-designed, easy to use, and fast.
- Working hard to have a reliable system for processing terabytes of data on the server-side.

Python Client & ML Integrations:

- Developing our Python API.
- Connecting our APIs with many more and less popular machine learning frameworks.
- Researching ways of integrating and doing this effectively.
- Open-source.

Web UI:

- Taking care of the user panel available through the web user interface.
- Creating rich data visualization (interactive charts, tables, and more).
- Implementing a modern, well-performing front-end.

Site Reliability Engineering:

- Putting services reliability, security, and high availability in the first place.
- Responsible for productionisation, observability, releases, and stability.
- Optimizing the development process and tooling.

- The Engineering Team works in **Kanban**, with **flexibility** in choosing tasks.

- Each task is specified by Software Engineers and Product team together.

- The Engineering Team spends most of its capacity on **developing new features**.

- Some time is also spent on **maintenance** (issues investigation, fixing, deployment, controlling technical debt).

- Within working hours, we have one person from the Core Systems and one from the Web UI responsible for **issues investigation** to **minimize context switches for others.**

- Additionally, we have **24/7 on-call duty for emergencies** (based on automated monitoring), **extra paid.**

Web UI:

• **JavaScript / TypeScript**
• **React**
• **Redux**
• Swagger
• Sentry
• Domain-Driven Design
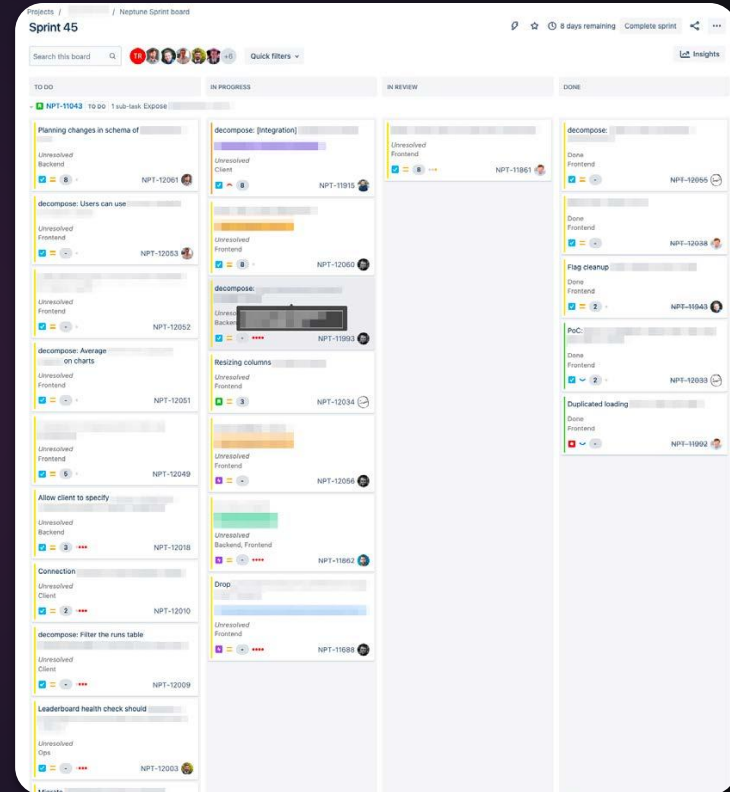
Backend:

• **Rust**
• Java & Spring, Scala, Kotlin (parts)
• Kafka
• Clickhouse
• MySQL
• Protobufs
• Swagger

Infrastructure, deployment:

• **Google Cloud Platform**
• **Azure**
• **Kubernetes**
• Terraform
• Helm
• Gitlab CI/CD

Python Client:

• **Python API**
• **OpenAPI, Swagger**
• **Protobufs**
• Integrations with top machine learning frameworks and tools: Jupyter Notebook, TensorFlow, Keras, PyTorch, R, Scikit-Learn, fastai, and more

Do you need any machine learning skills?

As mentioned at the beginning – we hire great Software Engineers to develop an industry-standard system for ML Engineers.
98% of the time, we use "classic" software engineering skills to develop a highly scalable system.

The Dev Team does not do machine learning – our customers do.

You do not need to be a data scientist to join our team. But you will have the opportunity to learn how machine learning is done in production at scale.

Challenges

Neptune is a complex product with many customers and a high load. Ideas for the new features keep flowing constantly.

This means **tons of challenges.**

Take a look at the following few slides to learn about some that we had in the past, have currently, or will have soon.

All of these are from the **Dev Team's perspective**.

Long and wide data series

In neptune.ai, we have runs that can contain many attributes — and many series items within them.

We were prepared for hundreds of parameters (think about attributes, how well the model behaves, infrastructure utilization, etc.) and millions or more records within them.

But... some customers created runs with **hundreds of thousands** of different parameters, and we faced challenges: *how to show table-wide for hundreds of thousands of columns, how to process this effectively*?

Should we limit it? **Not so easy.**

Front-end maintainability

In the beginning, we had just a few software engineers. The front-end that was created was a kind of monolith. Time flew, more and more software engineers joined, and functionalities grew quickly.

Challenge: *how to keep the maintainability of the front-end?*
We talked about it and decided to do **a major (iterative) re-architecture** to have a components-based, domain-driven design.

On-prem vs. SaaS

Neptune has two kinds of deployments:

1. The main SaaS deployment that is fully maintained by our team;

2. And on-prem deployments that are done by some of our larger customers.

Challenges: *how to develop software and choose technologies to **make the deployment generic enough** so that it can be done **effectively** both on our side and on the side of many customers with different infrastructures?*

Terabytes of data

Neptune grows. Our customers produce gigabytes of data. Gigabytes sum up to terabytes. Data of different kinds – floats, strings, images, and different objects.

It is definitely non-trivial to store and process this effectively, with response times measured in *tens of milliseconds*.

We went through different improvements like introducing **Elasticsearch or S3 on top of MySQL.**

As a result, **we are really fast!**

New generic features and backward compatibility

Constant challenge: *how to **keep adding features** requested by our users and make them generic enough to be useful in many different use cases, and how to grow and **keep backward compatibility***?

The Dev Team works closely with the Product Team to design new back- and front-end features, solving generalized use cases. And we make sure to design new functionalities in a way that will not be disturbing to our current customers.

Why should you join?

In neptune.ai, **we don't guard team boundaries**.

It means anyone can suggest an idea for a new great marketing campaign or propose new features to develop.

All product ideas, backlogs, and plans are **available for everyone at Neptune** to view and contribute, and the Dev Team works actively on new features together with the Product Team.

As a company, we grow steadily and plan to continue doing that. That means a bigger product over time and more people engaged.

You will have plenty of opportunities to grow. You will learn how to build modern software at scale. You can propose and lead introducing new technologies to our stack. You may also have the opportunity to own a smaller or bigger area or lead a team.

You will have an opportunity to work with a great team full of people who are open to **learning from each other**.

We not only like to work together but simply **enjoy each other's company**.

Neptune's success is your success.

We offer **Employee Stock Option Plan (ESOP)**.

Having this, you may become a co-owner and directly participate in Neptune's success.

Read more on how ESOP works:
https://bit.ly/3NssOb9

Perks and benefits

⇄ Indefinite B2B contract

$ 20 paid service-free days

▪ Employee Stock Option Plan (ESOP)

🏠 Flexibility of work — 100% remote with an office in Warsaw and coworking spaces in Wrocław, Poznań and Kraków available

📄 Amazon Books company library

👤 $50 per month for learning and development

★ Regular team building events

◉ Co-financing of medical care and sports card (currently only available for people residing in Poland)

Recruitment process

Our recruitment process is **fully remote**, and we make our best to keep it **short and effective**. Sometimes, we adjust our approach depending on the position and its complexity.

We value your time, and our goal is to shorten the recruitment process to two weeks, assuming we can meet each other's availability.

CV review

We need 1-3 days to review your CV and decide whether we invite you to the process. A human being screens every CV, and we try to ensure you always get an answer. If you didn't, reach out!

HR screen

Video call with your Recruiter – we dive into your experience and tell you more about the role and neptune.ai. This meeting takes about 30-45 minutes. Use this time to **ask questions and learn more about Neptune.**

All our calls are recorded for internal purposes, subject to your acceptance. We never share the recordings externally.

Technical assessment

If the feedback is positive, we proceed to a technical assignment first and then (assuming a positive decision) the video call technical interview. It takes about 2.5 hours, and you can expect deeper technical questions and some elements of live coding. Typically, it's done by two senior engineers from our side.

Interview with Manager

After the tech interview, you will have a chance to speak with your future manager. This interview will take about an hour and might be conducted by our CTO or the Engineering Manager. You can expect deeper, position-specific questions.

Reference check

Before we make a final decision, we will ask you to provide us with references to people we can reach out to for a brief chat, e.g., your previous manager or peers.

Offer/decision

We act quickly – after the reference check, we will need one working day to come back with the decision.

If it's positive, we will present our offer.

Here are a few hints:

- **Intro call (HR screen) is to get to know each other** – ask us questions about the company; you do not need to prepare for this stage.

- **You do not need ML skills** – it is not the core of the Dev Team.

- **For the tech interview**, **prepare the IDE**, the dev environment of your choice, to be ready to participate in the **live coding**.

- **Do not get stressed after mistakes** – these happen and may not matter in the end.

- **Think aloud** to help us understand how you solve tasks!

Optional quick screening questions examples:

• What are the differences between list and map?

• Describe HTTP protocol in a few sentences.

Interview questions examples (may vary between levels):

• What is constructor overloading?

• [Frontend] How do classes look like in TypeScript?

• [Live coding] Implement simple crawler downloading HTMLs tree up to 100 subpages of a given site.

• [For senior+] Design large scale simple analytics system (guests online, unique visitors per month).

Looking for more?

Neptune's YT Channel
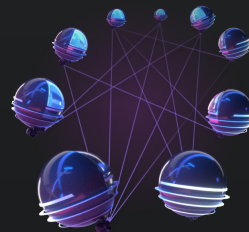
Neptune Blog

Join us!

Check current openings:
https://neptune.ai/jobs

Join us and let's celebrate success together!

Questions?
jobs@neptune.ai