

Monoracle

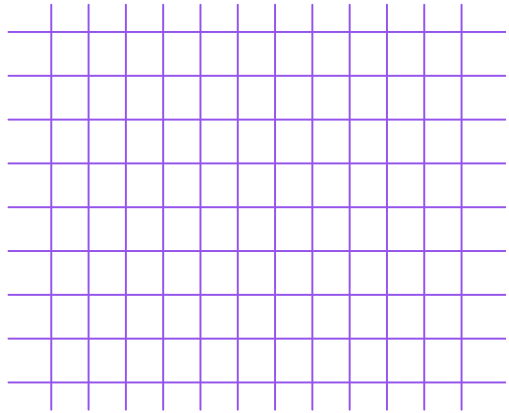
---

# Api to oracle in seconds!

→ [monoracle.xyz](https://monoracle.xyz)



Monoracle



---

# Problem

Blockchains are isolated systems that can't access external data on their own. Traditional oracle solutions require:

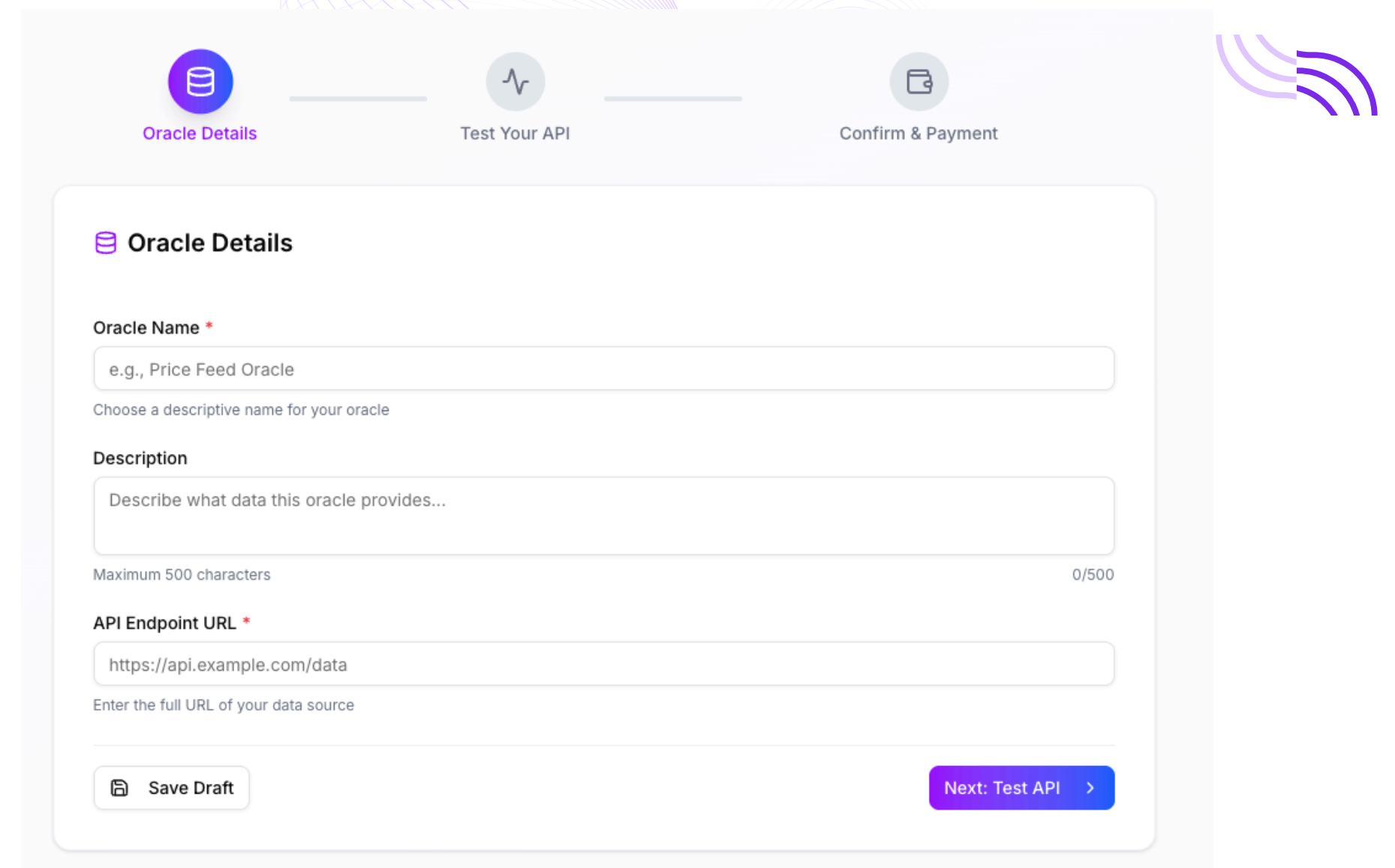
- **Expensive oracle service subscriptions**
- **Manual data feeding infrastructure**
- **Technical blockchain expertise**

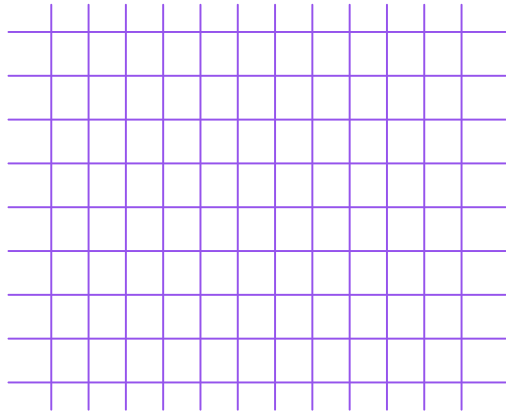


**Monoracle**

# What we built

- No-Code Oracle Creation with just simple api call!

The image shows a web interface for creating an oracle. At the top, there's a progress bar with three steps: "Oracle Details" (active, with a database icon), "Test Your API" (with a pulse icon), and "Confirm & Payment" (with a wallet icon). The "Oracle Details" section contains three input fields: "Oracle Name" with a placeholder "e.g., Price Feed Oracle" and a hint "Choose a descriptive name for your oracle"; "Description" with a placeholder "Describe what data this oracle provides..." and a character count "0/500" (with a hint "Maximum 500 characters"); and "API Endpoint URL" with a placeholder "https://api.example.com/data" and a hint "Enter the full URL of your data source". At the bottom, there are two buttons: "Save Draft" and "Next: Test API >". The interface is light gray with purple accents.



# How It Works

Our no-code oracle platform transforms any public API into blockchain-ready data.

Multiple nodes fetch and verify information in parallel, reach consensus on the result, and deliver the verified data on-chain — all automatically, without writing a single line of code.

- **User Defines the Data Source (No Code)**
- **Oracle Platform Generates a Fetch Task**
- **Get data and use on your frontend.**



**Monoracle**



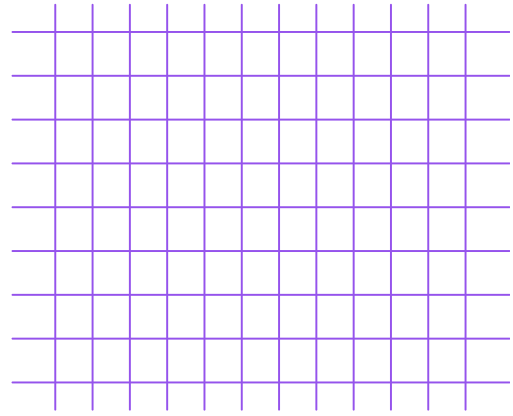
---

# monoracle-sdk

- Fully on-chain toolkit
- Verified oracle data
- Real-time access
- No centralized APIs
- **View documentation:** <https://monoracle.xyz/docs>
- **View on npm:** <https://www.npmjs.com/package/monoracle-sdk>



Monoracle



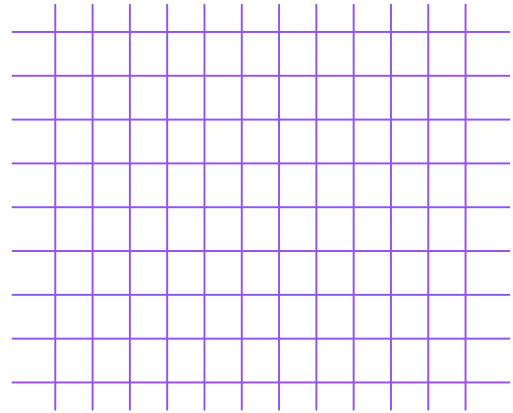
---

# monoracle marketplace

- Decentralized data marketplace
- Verified community datasets
- Transparent contributor rewards
- On-chain knowledge sharing



Monoracle



---

# ✦ Use cases

**Monoracle in logistics:** <https://monoracle.xyz/use-cases/demo-1>

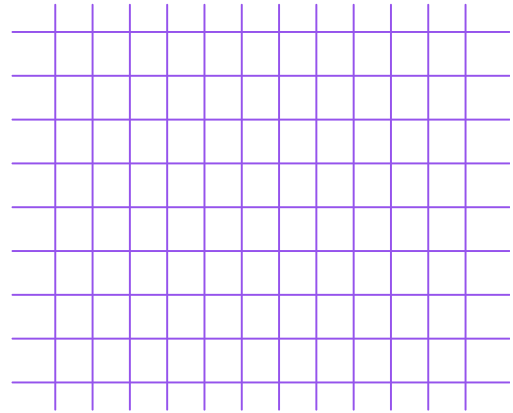
**Monoracle in sports:** <https://monoracle.xyz/use-cases/demo-2>

**Monoracle integration with any api:**  
<https://monoracle.xyz/use-cases/demo-3>



Monoracle





## Differences between Chainlink

### Monoracle

SDK-first approach lets developers easily display verified on-chain data in frontend apps.

User-friendly, plug-and-play datasets and on-chain jobs make development easy and fully decentralized.

Marketplace lets the community share and discover datasets, boosting collaboration.

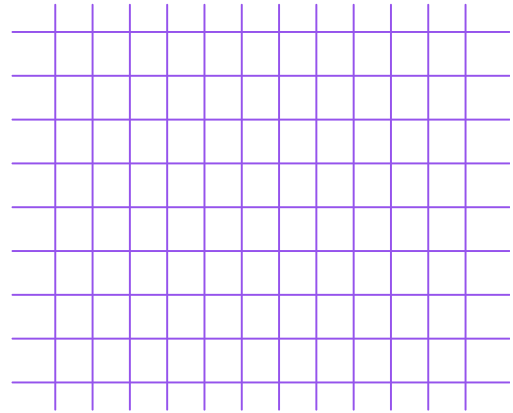
### Chainlink

Designed primarily for smart contracts

Frontend integration usually requires extra backend work.

Limited community dataset sharing.





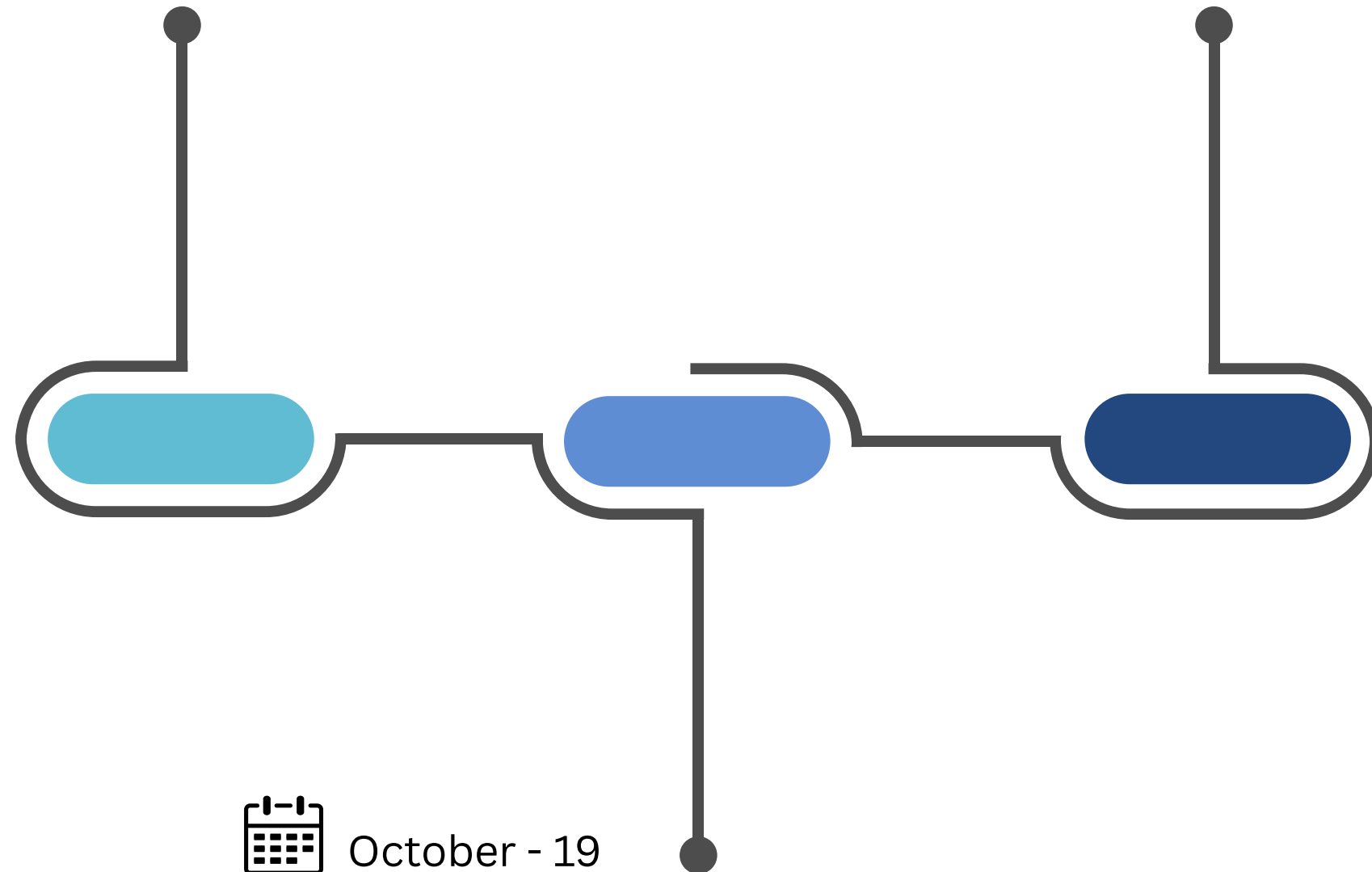
# Timeline



October - 4

**We started monoracle project  
at Buildermare Monad Blitz Hackathon**

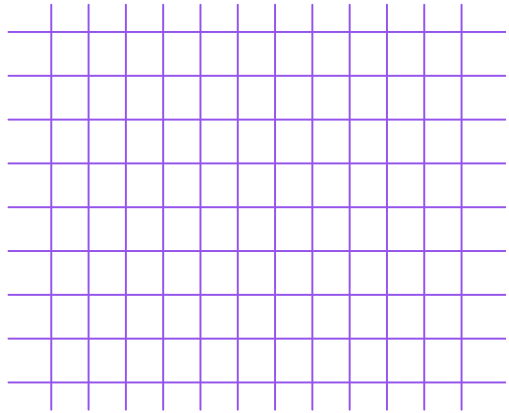
**Currently working on new features  
Eg: Marketplace**



October - 19

**We deployed Monoracle on production,  
testing on testnet.**





# Team



## Akifcan Kara

Fullstack developer, web3 developer

[github.com/akifcan](https://github.com/akifcan)

[www.linkedin.com/in/akifcan-kara](https://www.linkedin.com/in/akifcan-kara)



Monoracle

# Thank you :)

Thank you for your attention! We look forward to collaborating.  
Please reach out with any questions.

- [akfkara97@gmail.com](mailto:akfkara97@gmail.com)
- [https://x.com/kara\\_akifcan](https://x.com/kara_akifcan)

