Ups and Downs: Onboarding a Million Users to Layer-2

Matt Pearring
THE STATE OF ARBITRUM

$2.5B+
TOTAL VALUE LOCKED

51%+
ROLLUP MARKET SHARE

400+
DAPP INTEGRATIONS

1.4M+
UNIQUE ADDRESSES

BLUE CHIPS
- UNISWAP
- SUSHISWAP
- AAVE
- CURVE
- YEARN

NATIVE
- DOPEX
- VESTA
- SPERAX
- GMX
- TRACER

INFRASTRUCTURE
- CHAINLINK
- SAFE
- INFURA
- THE GRAPH
- ALCHEMY
- ETHERSCAN
A FEW AREAS OF FOCUS

BRIDGING UX FOR EVERYONE
CONTINUOUS COMPATIBILITY
SCALING, GENERALLY
ZERO DEGREES OF SEPARATION BRIDGING
The State of Bridging

Bridging assets continues to be challenging (and risky)
Bridging assets continues to be challenging (and risky)

UX for users is tricky (or impossible) depending on the asset
Bridging assets continues to be challenging (and risky)

UX for users is tricky (or impossible) depending on the asset

Not all token implementations are created equal
I.e. standard vs. custom bridging
Bridging assets continues to be challenging (and risky)

UX for users is tricky (or impossible) depending on the asset

Not all token implementations are created equal, i.e. standard vs. custom bridging

SOURCE: https://bridge.arbitrum.io
ZERO DEGREES OF SEPARATION

WHAT'S NEXT

Fast withdrawal confirmations* (enabled by Data Availability Committees)
ZERO DEGREES OF SEPARATION

WHAT'S NEXT

Fast withdrawal confirmations* (enabled by Data Availability Committees)

Deep wallet integration with fast bridging
**ZERO DEGREES OF SEPARATION**

**WHAT'S NEXT**

- Fast withdrawal confirmations* (enabled by Data Availability Committees)
- Deep wallet integration with fast bridging
- UX Iteration: Discovery of fast bridging, and more performant UIs
ZERO DEGREES OF SEPARATION

WHAT'S NEXT

- Fast withdrawal confirmations* (enabled by Data Availability Committees)
- UX Iteration: Discovery of fast bridging, and more performant UIs
- Deep wallet integration with fast bridging
- Bridge aggregation as standard functionality
SEAMLESS COMPATIBILITY

THE STATE OF TOOLING
On Arbitrum, drop-in compatibility is standard, all EVM programs work out of the box.
On Arbitrum, drop-in compatibility is standard, all EVM programs work out of the box.

Core infra across L1 is generally available on L2.
SEAMLESS COMPATIBILITY

THE STATE OF TOOLING

On Arbitrum, drop-in compatibility is standard, all EVM programs work out of the box.

Core infra across L1 is generally available on L2.

SDKs and frameworks are beginning to multiply.
On Arbitrum, drop-in compatibility is standard, all EVM programs work out of the box.

Core infra across L1 is generally available on L2.

SDKs and frameworks are beginning to multiply.

The State of Tooling

- Node Functionality
  - Arbos
  - Geth Core

Seamless Compatibility
SEAMLESS COMPATIBILITY

WHAT'S NEXT

Broader language support for smart contract execution
SEAMLESS COMPATIBILITY

WHAT'S NEXT

- Broader language support for smart contract execution
- Continuous improvement on gas pricing and estimation
SEAMLESS COMPATIBILITY

WHAT'S NEXT

- Broader language support for smart contract execution
- Continuous improvement on gas pricing and estimation
- Widely agreed upon standard dev frameworks
SEAMLESS COMPATIBILITY

WHAT'S NEXT

- Broader language support for smart contract execution
- Continuous improvement on gas pricing and estimation
- Widely agreed upon standard dev frameworks
- Long-term: more web2 -> web3 rails and rants
PERFORMANCE = ADOPTION

THE STATE OF SCALE
Teams are making strides in performance and throughput every week
Teams are making strides in performance and throughput every week.

Arbitrum Nitro shipped just last month, 7x’ing our capacity.
PERFORMANCE = ADOPTION

THE STATE OF SCALE

- Teams are making strides in performance and throughput every week
- Arbitrum Nitro shipped just last month, 7x’ing our capacity
- Efficiency is crucial across rollup systems from sequencing, through execution, to validation
PERFORMANCE = ADOPTION

Teams are making strides in performance and throughput every week.

Arbitrum Nitro shipped just last month, 7x'ing our capacity.

Efficiency is crucial across rollup systems from sequencing, through execution, to validation.

THE STATE OF SCALE

<table>
<thead>
<tr>
<th>Name</th>
<th>Send ETH</th>
<th>Swap tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metis Network △</td>
<td>&lt; $0.01</td>
<td>$0.05</td>
</tr>
<tr>
<td>Loopring</td>
<td>$0.03</td>
<td>$0.32</td>
</tr>
<tr>
<td>Arbitrum One</td>
<td>$0.03</td>
<td>$0.10</td>
</tr>
<tr>
<td>ZKSync</td>
<td>$0.04</td>
<td>$0.09</td>
</tr>
<tr>
<td>Optimism</td>
<td>$0.11</td>
<td>$0.17</td>
</tr>
<tr>
<td>Boba Network</td>
<td>$0.18</td>
<td>$0.31</td>
</tr>
<tr>
<td>Polygon Hermez</td>
<td>$0.25</td>
<td>-</td>
</tr>
<tr>
<td>Aztec Network</td>
<td>$0.28</td>
<td>-</td>
</tr>
<tr>
<td>Ethereum</td>
<td>$0.70</td>
<td>$3.49</td>
</tr>
</tbody>
</table>

SOURCE: https://l2fees.info
MANAGING GROWTH

DEALING WITH STATE BLOAT
Node architecture must continue to evolve
MANAGING GROWTH

DEALING WITH STATE BLOAT

Node architecture must continue to evolve

Client Optimization (Erigon, etc.)
MANAGING GROWTH

DEALING WITH STATE BLOAT

Node architecture must continue to evolve

Layer 2s and Layer 1 mainly dealing with the same set of problems / core bottleneck

Client Optimization (Erigon, etc.)
MANAGING GROWTH

DEALING WITH STATE BLOAT

Node architecture must continue to evolve

Layer 2s and Layer 1 mainly dealing with the same set of problems / core bottleneck

Client Optimization (Erigon, etc.)

Rentable Storage
MANAGING GROWTH

DEALING WITH STATE BLOAT

Node architecture must continue to evolve

Layer 2s and Layer 1 mainly dealing with the same set of problems / core bottleneck

Client Optimization (Erigon, etc.)

Rentable Storage

State Expiry & Regenesis
MANAGING GROWTH

GROWING STATE USAGE
MANAGING GROWTH

GROWING STATE USAGE

Emerging use cases, with lower barriers to entry
(i.e. Games with less pay-to-win)
MANAGING GROWTH

SCALING

GROWING STATE USAGE

Emerging use cases, with lower barriers to entry
(i.e. Games with less pay-to-win)

Continuous iteration of DeFi onboarding
MANAGING GROWTH

SCALING

GROWING STATE USAGE

- Emerging use cases, with lower barriers to entry
  (i.e. Games with less pay-to-win)
- Continuous iteration of DeFi onboarding
- Insurance as a core primitive