202405 - Darwinia Final Report

Darwinia Documentation Review by PaperMoon

Messaging Review

Summary

Messaging is confusing as the landing page does not reference anything about the different offerings you go into in the docs (MSG Port). Also, it seems the messaging is focused on cross-chain, but you don't clarify the Parachain's role there.

The landing page should have a clear call to action for developers, answering some of the following questions:

- What might they be looking for?
- What does Darwinia offer (concretely)?
- Information you want to highlight for Developers

High-level titles should indicate the path to developers, not the tech you are building. Paths should lead to the precise tech (check informational architecture).

Content should be structured in clear verticals. Our review shows you have two main offerings: MSG-Port and the Darwinia Parachain. Because they are so different, we recommend you create sub-sections for each main vertical; please check the Informational Architecture Review.

Navigation

The navigation bar should point developers in the right direction. The right question to answer is: *What is the developer looking for to get to the content he should find?* The high-level current form is:

- MSG Port:
  - What does this mean? What is *MSG Port* to a developer? Is it a cross-chain solution? Is it a smart contract solution? The correct title heading should answer these questions.
- EVM+:
  - What does this mean? How are you expecting the developer to click this page?
- RING:
  - A page should not contain only a single piece of content. Using heading titles more efficiently is recommended.
- Ecosystem Collaboration:
  - Is this a page you want developers to see in the developer documentation? A single page for a heading is not recommended.

General Content Organization

Content should be better structured. Our review finds that the current content structure is to "throw everything under the main vertical label." Subsections for such objects are crucial.

Note: typos should be minimized as they impact credibility. We recommend using language models or tools such as Grammarly or others to correct texts.

MSG-Port

Content flow has no structure. Topics abruptly change. For example, if we follow the general high-level content of each item menu on the left-hand side:

- Overview (conceptual)

- Glossary (conceptual)
- Msgport Interfaces (technical, introductory)
- Msgport Workflow (technical, introductory)
- Msgport Workflow (technical, tooling)
- Msgport Scan (technical, tooling)
- Messaging Protocols (conceptual)
- Supported Networks (technical, introductory)
- Use Cases (conceptual, lead to external pages like XToken)
- Tutorial (technical, practical examples)
- FAQ (conceptual)
- TroubleSome (Troubleshoot? no content)

Therefore, organizing the content in subsections is recommended.

Subsections should answer the following questions:

- What are the different aspects you want to show developers about MSG-Port? For example, subsections could be classified as:
  - Conceptual aspect: concepts, how the tech works at a conceptual level, conceptual use cases.
  - Technical aspect: supported network, getting started, how to configure the solution
  - Practical aspect: end-to-end tutorials that show how MSG-Port works
- What call to action do you want developers to do? Content should follow a concrete structure that you guide developers through and that has a logical order. For example, for a getting started section:
  - Supported Networks
  - Setup (getting tokens, whatever other things are needed)
  - MSG-Port Configuration
  - MSG-Port message building
  - MSG-Port message
  - What to do next?

EVM+

First, we recommend you build this page with the left-hand-side navigation menu tabs collapsed.

Similar comments as to the MSG-Port section. Overall, the content is all posted into a single subsection with the same title as the heading title. A mix of conceptual and technical content breaks the flow a developer might need when going through hefty documentation pages.

Also, the section does not follow the same general structure as MSG-Port. Developers welcome structured content. For example, Overview is at the top of the navigation menu, but Glosarry is second last, while in MSG-Port, it was second.

Subsections should be classified similarly to the structure mentioned for MSG-Port, dividing conceptual, technical, and practical aspects of the documentation.

A section on common Ethereum tools and Darwinia Network is missing, which is important so developers know how to set up common tools like Ethers, Hardhat, and Foundry, among others. Frontier unlocks all of this with no code changes, which is something that must be shared.

RING

Having a page dedicated to a single piece of content about the token is not good practice. Moreover, The *https://docs.darwinia.network* domain should be thought of as documentation for developers. Any page regarding the token should be either on the main domain page or it has to be specifically laid out for developers.

We recommend removing this page entirely

Ecosystem Collaboration

It is important that builders see how they can collaborate with Darwinia. Nevertheless, it is important to locate this content appropriately to coherently with the developer's message. For example, we put the Bug Bounty and Grants Program information on the landing page for Moonbeam.

We recommend moving this to a different section.

Proposed Informational Architecture

Darwinia Docs

One critical aspect to consider when organizing the content on your documentation site is the audience for which the content is intended. The following are some questions that you should ask yourself when creating and organizing content:

- Who is this content intended for? Is it for a developer, an infrastructure provider, a community member, etc.?
- What is the goal or objective of the content? Does it provide instructions, explain concepts, etc.?
- What are the key topics or sections to cover? Identify and prioritize the topics based on importance to the audience

When considering the above questions for the Build section, the answers might be as follows:

- The content is intended for developers
- The goal of the content is to provide instructions on how to build
- Key topics include: the essentials to start building (network configurations, RPC endpoints, testnet faucet, etc.), the tools that can be used to build (libraries, development environments, wallets), how to interact with the network’s features (JSON RPC, precompile contracts, Substrate pallets), etc.

When considering the above questions for the Learn section, the answers might be as follows:

- The content is intended for developers, non-technical stakeholders such as project managers or product owners, or researchers
- The goal of the content is to explain the fundamental concepts and principles and understand advanced topics
- Key topics include the network's capabilities (Ethereum compatibility, etc.), its architecture, its features, and how these features work (staking, XCM, governance, etc.).

Regarding the community documentation on the DAOs and the "how-to" guides that primarily target non-technical users and focus on tasks unrelated to the technical implementation or development aspects of the network, it may be worth considering whether they belong on the developer documentation site or the main network site. Keeping these on the main network would offer higher visibility and make more logical sense.

General feedback

- Always keep your target audience in mind
- There should generally be a rationale behind content placement within documentation. Content can be organized in a logical manner that follows a clear hierarchy; it can be organized based on a workflow, or if there is no hierarchy or workflow, then content can be organized alphabetically
- Anticipate how your documentation might grow over time and possible future pain points.
    - As you add more documentation, are you going to overcrowd the navigation?
    - Can you prepare for the growth now? Moving content means that you should be adding redirects for moved content. Redirects are important for maintaining the integrity of your website’s URL structure, preserving SEO value (findability of your documentation on search engines), and providing a positive user experience (no broken links). How can you avoid having a long list of redirects by planning ahead for the future?
- All content should be organized into subsections for clarity and coherence within the main sections. For instance, consider the Learn Darwinia section, where the Wallets, Governance, and Collator Staking pages may be perceived as part of the Chains section due to their placement on the UI. To enhance visual organization and user experience, it's recommended to group the Wallets, Governance, and Collator Staking pages under a designated subsection (not necessarily together under one subsection, as the grouping needs to make sense), ensuring a more cohesive and intuitive navigation flow

- Titles of sections and pages should clearly convey the content and purpose of the section or page.

Build

(No need to add “on Darwinia” to the title anymore)

- Get Started
  - Networks
    - Overview
    - Darwinia
    - Crab
    - Pangolin
    - Rollup Testnet
    - Development Node
      - Run a Development Node
      - Development Accounts (to provide a quick reference)
  - RPC Endpoints (increases visibility)
  - Block Explorers (increases visibility)
  - Testnet Faucet
  - Wallets
- Development Tools
  - Ethereum Tools
    - Web3 Libraries
      - Ethers
      - Web3
      - <Additional libraries?>
    - Development Environments
      - Hardhat
      - Foundry
      - <Additional development environments?>
    - Verify Contracts
  - Substrate Tools
    - Polkadot.js API
    - <Additional substrate tools?>
  - Protocol Integrations (If you intend to add other integrations, like Oracles, etc., this format allows you to continue adding more sections)
    - Indexers
      - SubQuery
    - <Additional integrations?>
- Precompiled Contracts (All of these pages should include the addresses of these precompiles, an example of how to use the precompile, the actual solidity contract, and ideally an overview of the functions the precompile provides)
  - Overview
  - Asset
    - KTON
    - USDT
  - Governance (While currently consisting of only one page, including this section ensures consistency with the structure of the rest of the precompiles section and it enhances clarity)

        - Conviction Voting
    - Staking
        - Staking (Saw this note and I'm curious what the long-term plan is? "If you want to learn how to interact with these Collator Staking module, pleast vist the Staking Precompile Page, be aware that this contract may be deprecated in future versions.") - Note the typos in this line; this was copied directly from your docs
        - Deposit
    - Utility
        - Ethereum Mainnet
        - Dispatch
        - State Storage
- Module APIs
    - Deposit Module - Leads to a 404; couldn't find its new home
    - Staking Module - Leads to a 404; couldn't find its new home
    - <Additional modules?>

Network Operators

(Note: we moved the development node guide because it is targeted towards developers, as it provides them with their own development environment, similar to what Hardhat Network provides. Whereas the rest of the guides are targeting infrastructure providers.)

- Get Started (add requirements, default ports, etc.)
- Database Snapshots
- Run an Archive Node
- Run a Collator Node
- Run a Tracing Node
- Run a Testnet Relay Chain Node - Leads to a 404; couldn't find its new home

Learn

(No need to add "Darwinia" to the title anymore)

- What is Darwinia? (Provide an overview of Darwinia)
- Architecture (Add information about Darwinia's architecture)
- EVM Compatibility
    - EVM+ Overview
    - Account System
    - Gas Compatibility
- Native Interoperability (Add XCM info. Could bridge the gap a bit with Msgport here, too)
- Staking (Add overview about the staking process, what does it look like, how are rewards calculated or disbursed. There are several variations of PoS, so you should clearly define what the staking process looks like on your network. The Commitment Token info page now leads to a 404, so I forget exactly what was on that page, but what is the Commitment Token? How does it play a role in staking? All these questions should be addressed)
- Governance
- FAQ

Community

(Note: This section is better off under the main darwinia.network site, as that is the website most token holders will be visiting. If we were not developers, we would not consider going to the developer documentation to find information on the community DAOs or how to stake them. The docs under Community Guides are currently in the Build section but should be separated because their target audience is token holders participating in on-chain activities. They are not targeting developers, as these are not guides that focus on programmatically transferring

assets or how to stake or submit a proposal programmatically. They also don't belong under the Learn section because they are not conceptual; they are how-to guides.)

- Overview
- Community Organizations
  - RingDAO
  - DCDAO
  - KtonDAO
- Community Guides
  - Accounts
    - How to Create an Account
    - How to Create a Multisig
    - Account Migrations from Darwinia 1.0 (Darwinia Migrations requires you to click on it to know exactly what content is in that section, this provides a little more context)
      - How to Migrate Existing Accounts
      - How to Migrate Existing Multisigs
  - Token Transfers
    - How to Transfer Native Tokens
    - How to Transfer USDT
  - Staking (although one item, put it into its own section for consistency)
    - How to Stake
  - Governance
    - How to Submit a Governance Proposal
    - How to Vote on a Proposal
    - How to Create a DAO

Msgport Docs

Again, you have to think about the target audience and the goals of the section. The Learn section should only include conceptual information, such as the underlying mechanics of Msgport. The build section should provide everything a developer needs to know to use Msgport, such as the interfaces and the tools available (message scan).

With these documents in particular, it's important to remember that the titles of sections and pages should clearly convey the content and purpose of the section or page. In the current structure, titles like "Hello Message Using Remix" could be renamed "Send a Cross-Chain Message."

Use cases are not developer tools, they're not how-to guides that show developers how to interact with Msgport, so they should not be under the Build section. There is no building involved with these pages.

All of the tools required to build, such as knowing if a network is supported, the cross-chain explorer, the interfaces, the API, etc., are needed for the development process; as such, they should go under the Build section.

Learn

- What is Msgport
- Supported Protocols
  - Overview
  - ORMP
  - XCMP
  - LCMP
- Use Cases

    - Order Clearing
    - Cross-Chain Token Bridge
    - Cross-Chain DAOs
    - Cross-Chain Account Abstraction
- Glossary (This should not be the first page in the section, it should be lower in the list as it should play a supporting role, not a lead role)
- FAQ

Start Building with Msgport

- Supported Networks (Extract info from GitHub and add to doc)
- Cross-Chain Explorer
- Msgport Interfaces
- Msgport Workflow
- Msgport API

Tutorials

- How to Send a Cross-Chain Message (Combine w/ script demo page. Or if you want to keep them separate, make sure that the script demo is actually a tutorial with step by step instructions. It should be more than just a link to an example repository)

Semrush (Crawler)

Organic traffic to the documentation site is extremely low. The estimated number of monthly visits that your target gets from organic search since January 2024 is as follows:

- January: 0
- February: 2
- March: 6
- April (as of the 16th): 0

The documentation site ranks for four keywords with the following positions:

- Blockchain indexer: 88
    - URL: https://docs.darwinia.network/indexer-guides-d4ca84582e124d18859ebaffc0804fff (results in 404)
- Dwellir: 67
    - URL: https://docs.darwinia.network/darwinia-db787b70edd14545a252122d1c7651d1 (results in 404)
- LCMP: 40
    - URL: https://docs.darwinia.network/lcmp-overview-d205c8d8e5794c789065aabdbb5f78c8 (results in 404)
- Darwinia: 21
    - URL: Darwinia Docs

Observations:

- There are 8 broken links on the documentation site that should be fixed
- Based on the above URL ranking, I assume these pages all moved recently, and the documentation site hasn’t been indexed. Once the site gets crawled again, all of these rankings will go away
- There are 100+ backlinks that are broken, resulting in 404s. Looking at this stat, it doesn’t appear that a system is in place for redirects. Highly recommend adding a redirect anytime a page is removed or relocated to preserve backlinks and maintain SEO ratings
- Couldn’t find a sitemap. If you don’t already have one, creating one and manually submitting it to Google is recommended. This can be a proactive step to improve the visibility of the site in Google's search results
- Improve page titles and add meta-descriptions to boost SEO

Lighthouse Testing

Lab data from Lighthouse suggests that the performance of the documentation site is in great shape.

- Performance: 98% (82% on mobile)
- Accessibility: 91%
- Best practices: 96%
- SEO: 90%

Some minor improvements can be made to boost the scores for the accessibility and best practices categories to 100%.

Unfortunately, not enough traffic is coming to the site to generate field reports of what real users are actually experiencing.

Conclusions

After thoroughly reviewing the documentation site, we can provide the following conclusions:

- Overall, messaging is extremely confusing. It is unclear what developers are looking for and how developers should navigate the developer documentation
- Darwinia is trying to document two extremely separate products under the same domain. This can be quite confusing for developers, as it complicates the overall message. For example, is Msgport related to Darwinia blockchain at all? How is Msgport related to EVM+? Separating each product might help concentrate efforts on each product's messaging
- Sections have no clear developer target and no clear structure between philosophical content, technical content, and demo/tutorials
- Keyword ranking is important for SEO impact, creating a concise list of keywords that are always repeated throughout articles is important to generate SEO traction
- Having 404s pages negatively impacts SEO scores, solving those should be of high importance
- Having tutorials with popular developer tools can help generate traffic to the website. Current traffic is pretty much non-existent. Keywords are important to start generating organic traffic from search engines
- In general terms, tutorials should always be working. If a developer follows a tutorial and runs into problems it will just encourage them to try something else that is not using your product