RegionX

1. General Project Information

Short description: *RegionX is a project designed to reduce deployment and operational costs for projects on the Polkadot network. It will enable existing parachains to procure only the Coretime they require, thereby lowering their operational expenses. It will also enable new teams aiming to deploy on Polkadot to do so at significantly lower expenses.*
Project Category / Type: *Software development*
Proponent: 12icMZXL8aMbKXC84XD4ASC4y22Zywi7ETgrksX7LZdjZwFA
Requested allocation: USD$165000 | The amount of DOT will be determined on the day of submission of the proposal based on the EMA-30 price(Subscan).
Discussion date: 16.3.2024
Governance referenda origin call: Medium Spender
Previous treasury proposals: *N/A*
White paper: https://github.com/RegionX-Labs/Docs/blob/main/RegionX-Core.pdf
Forum discussion: https://forum.polkadot.network/t/building-a-coretime-marketplace/6661

2. Context

Polkadot stands out as a service provider of trustless and resilient computation. Its consensus system is designed with validators grouped into cores, enabling the network to conduct multiple operations in parallel without compromising security.

Applications that intend to run on Polkadot need to reserve time on these cores. We refer to this time as ‘Coretime’. Polkadot makes very few assumptions about the tasks executed on cores, allowing for a great diversity of projects. However, due to this diversity, not all projects require the same amount of computational resources. In the current slot auction model, the option for Coretime procurement is very limited. The only option offered is access to an entire Polkadot core for a long period of time.

The new Agile Coretime model will allow for more flexible procurement methods. Although it provides the architectural foundation for flexible procurement, the options it offers are still fairly limited by itself.

Projects like RegionX will build on the foundation that the new model offers to enable greater flexibility in Coretime procurement.

3. Problem

The Agile Coretime RFC introduces two new allocation methods for Coretime on the Polkadot Ubiquitous Computer:

1. **Bulk Coretime:** Allocated in advance, this method involves periodic scheduling of Coretime through a specialized Coretime chain. It is designed for users who plan their Coretime needs ahead of time.

2. **Instantaneous Coretime:** In contrast, this method allows for allocation of Coretime on the Relay chain on an as-needed basis, right before usage and follows a block-by-block approach. It suits users requiring Coretime on short notice or for immediate tasks.

In this model, Bulk Coretime sold on the Coretime chain is represented in the form of NFT tokens, known as 'regions'. These tokens represent ownership of Coretime for a predetermined duration, allowing their owners to assign them to particular tasks.

Every 28 days, the specialized Coretime chain launches a new sales cycle known as the, 'Bulk Sale', for these regions, thus enabling users to secure them for the upcoming time period. This periodic sales mechanism ensures that users can plan for and access the necessary Coretime in advance.
The assigned tasks will be executable on a Polkadot Core once the region becomes active in the upcoming 28-day cycle.

Coretime can be acquired either directly from the Coretime chain or from the relay chain for Instantaneous Coretime. Since Instantaneous Coretime is very short-lived, it cannot be traded. However, Coretime purchased from the Coretime chain, being represented as transferable NFTs, makes the creation of a secondary marketplace both feasible and practical.

The need for a secondary marketplace arises from two main limitations associated with buying regions directly from the Coretime chain:

1. The Coretime chain exclusively offers regions representing ownership of an entire Polkadot Core over a 28-day period. This allocation is excessive for most projects, leading to inefficient usage of Coretime.

2. Regions acquired from the Coretime chain become active only at the start of the upcoming 28-day cycle. This limitation implies that any task assigned to such a region can only be scheduled on a Polkadot Core during this specified period. Consequently, projects in need of immediate or imminent Coretime access must opt for Instantaneous Coretime.

The RegionX secondary marketplace aims to address these issues by providing a platform where regions purchased from the Coretime chain can be further traded and exchanged.

Unlike the regions offered by the Coretime chain, the marketplace will feature regions with a variety of properties. This diversity is made possible because users can modify the regions before listing them for sale, utilizing the partitioning and interlacing methods available on the Coretime chain.

Additionally, in the secondary market, it will be possible not only to acquire regions for the upcoming sales period but also to purchase regions that are currently active.

This will offer teams seeking to deploy on Polkadot a significantly broader range of options for Coretime procurement.

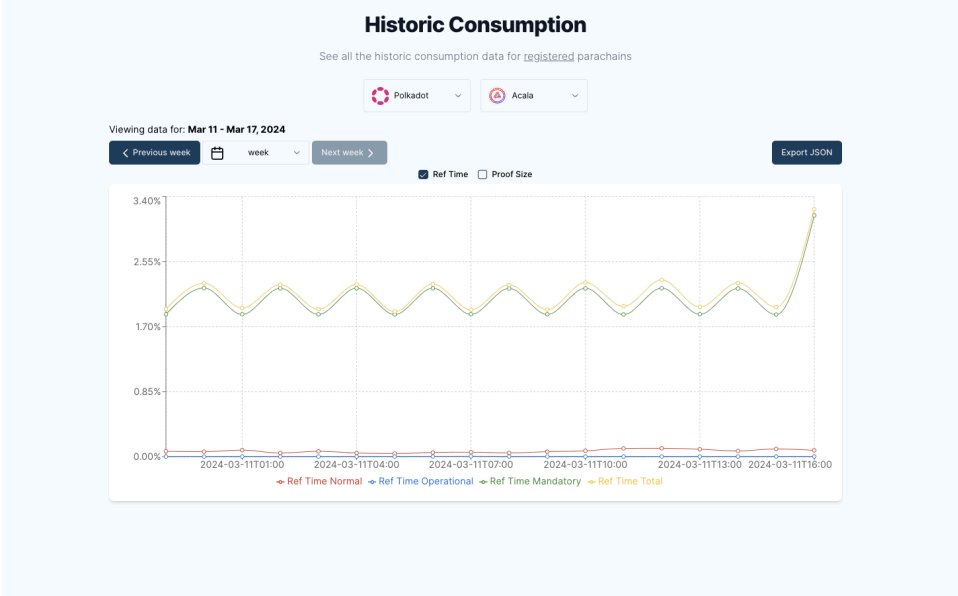
What does the data show?

We previously developed the Polkadot-Weigher tool, which monitors the Coretime consumption of existing parachains on Polkadot. Having operated for a while, the tool has now accumulated sufficient data, providing insights into the actual Coretime usage by projects on Polkadot.

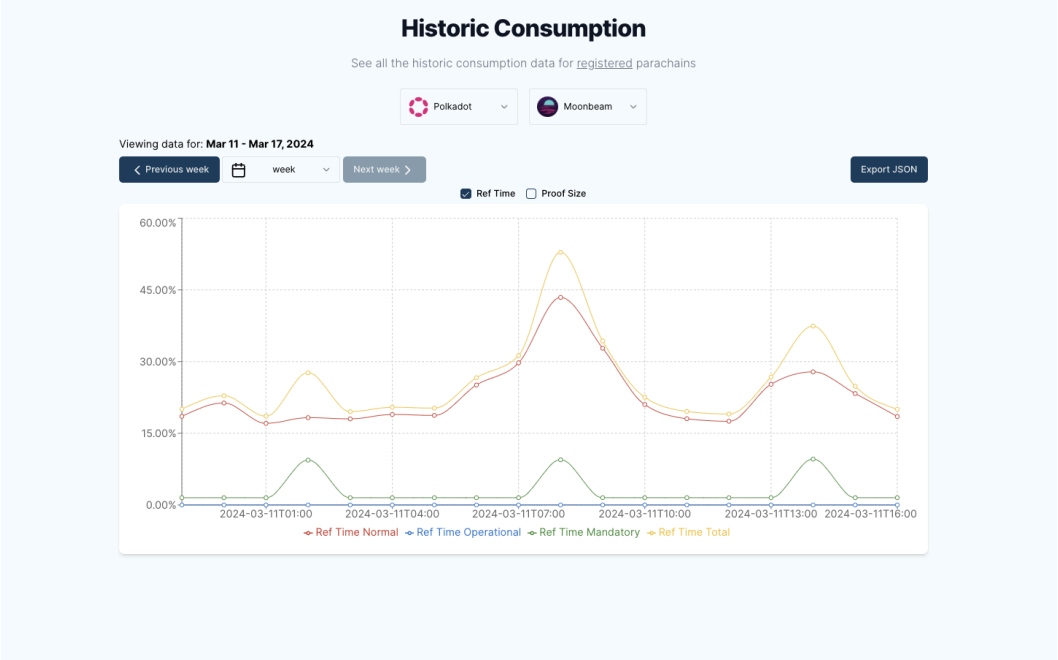
The tool demonstrates that even the well-known parachains on Polkadot do not require the full capacity of a Polkadot Core

Below are graphs depicting the Coretime consumption of several parachains over the past week: (The data can be verified by visiting the website: https://polkadot-weigher.com/history)

Acala averages 2-3% in Coretime usage:



Moonbeam averages 20-60% in Coretime usage:



Astar averages 2-6% in Coretime usage:

4. Proposal

We aim to create a marketplace designed to facilitate the trading of Coretime. Our goal is to offer a platform that existing parachains can utilize to procure the Coretime they require, thereby significantly reducing their operational costs.

However, our focus extends beyond just that. We aim to significantly reduce deployment costs on Polkadot, enabling more teams to bring their projects and ideas to production.

Since the Coretime chain doesn't support contract deployment, the marketplace logic needs to be implemented on another chain. Given that Coretime is represented in the form of NFTs, it's possible to transfer these assets to a chain that can host the marketplace. This is why we've invested significant time in creating a solution to transfer Regions using XCM.

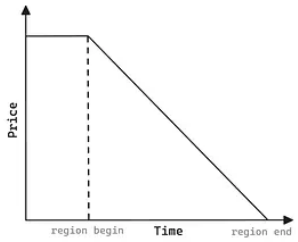
For easier understanding, we've divided our project into three main components:

1. Core of the Market

The core of the marketplace works based on the order book model as outlined in our white paper. Coretime regions put up for sale are categorized into two types:

- **Active regions.** The tasks that are assigned to active regions can currently be performed on a Polkadot core.
- **Inactive regions.** These are the regions that will become active in the upcoming Bulk period. The regions purchased from the Coretime chain fall into this category until the start of the next Bulk period.

The pricing of inactive regions remains fixed at the amount initially determined by the seller. In contrast, the cost of active regions follows our dynamic pricing model, which accounts for gradual depreciation in value. Accordingly, the pricing of these regions reflects their decreasing value over time.



As part of our W3F grant, we developed the core component of the marketplace in the form of ink! contracts. Given our aim to integrate this component into the RegionX runtime, we will be migrating it to a FRAME pallet.

2. Order Processor

Purchasing Coretime from the secondary market through a user interface is a practical approach when the parachain is not yet operational, necessitating manual acquisition of Coretime by those intending to deploy it. This means that all new teams planning to launch on Polkadot will initially obtain Coretime manually via a UI interface.

However, parachains that are already up and running on Polkadot shouldn't keep depending on the initial team for Coretime. For a parachain to be truly decentralized and resilient, it needs to establish a mechanism to autonomously procure Coretime, independent of any specific individuals.

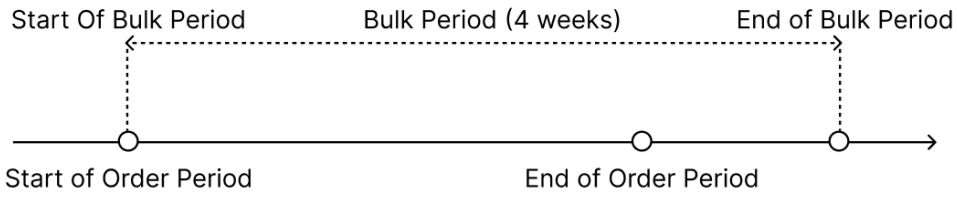
Additionally, it's crucial to acknowledge that parachains should not programmatically preset their market actions. Due to the fully transparent nature of these systems, such predetermined actions could easily lead to front-running attacks.

Parachains currently active on Polkadot must procure 'inactive' Coretime regions on a monthly basis for the upcoming Bulk period. By doing so, they ensure guaranteed access to Coretime in advance, maintaining uninterrupted operation of the chain.

In the order-based model we propose, each parachain can define the specific properties of the region it is looking for. For instance, a parachain might require a minimum of 50% of a Polkadot Core. Such requirements could be adjustable through the parachain's on-chain governance system.

At the beginning of every bulk sale period, each parachain places an order on the secondary market, specifying their Coretime needs. These orders initially start with a bidding amount of zero. Anyone can support these orders by contributing their DOT tokens. This system is akin to the parachain slot auction crowdloan model, where community contributions help secure the parachain's slot in the Polkadot network.

At the end of the order period, which is shorter than the bulk period, no new orders can be placed. The existing orders are then processed against the available regions for sale, with orders that have gathered the most DOT token contributions receiving the 'best deals' on the market.



A parachain may choose not to depend on crowdfunding to secure Coretime. If the parachain possesses DOT tokens it will be able to support its own order. However, in this scenario, the parachain's governance system must reach a consensus on the amount of tokens allocated for Coretime procurement.

3. Reward System

Just as participants in a crowdloan receive rewards if the parachain secures a slot, the parachain can determine the number of tokens allocated for rewarding users who contributed to the order. This mechanism incentivizes community support by offering tangible returns for contributions.

Moreover, this rewarding mechanism is practical because most parachains do not possess DOT tokens. Thus, they can motivate users to acquire Coretime on their behalf, offering rewards in the parachain’s native currency.

The amount of tokens allocated to rewarding contributors can be decided by the parachain’s governance system. These rewards could either be newly minted tokens, potentially leading to inflation, or sourced from the parachain’s treasury. It is up to the parachains to decide.

Coretime Traders

In the system we are building, we expect to attract users interested in making a profit by trading Coretime. A competitive environment of Coretime traders will significantly benefit the network, as they compete to make the best offers for Coretime procurement. This competition will be highly advantageous for those looking to purchase Coretime.

We are sure that there will be a variety of interesting strategies and trader bots developed to automate this task. These Coretime traders will strive to determine the most efficient allocation of Coretime, aiming to accommodate as many parachains as possible with the minimum amount of Cores purchased to maximize their profits.

4.1. Available solutions

There are currently no projects within the ecosystem that offer Coretime trading capabilities. While Lastic has announced plans to develop a Coretime marketplace, the absence of documentation on their marketplace's functionality prevents us from making a comparison.

After discussions with Lastic, we would be open to permitting them to deploy their project in the form of smart contracts on the RegionX parachain, enabling them to leverage our infrastructure. However, no official agreement has been made yet, and we are still in the phase of discussing this topic.

4.2. Scope of work

Our plan is to split the development into two phases. The first phase, for which this proposal is written, is for the development of the components that will be part of the RegionX parachain. This phase will cover all the fundamental components necessary for creating a Coretime marketplace.

The second phase involves developing modules that will facilitate integration with the marketplace for existing parachains. Having standardized and highly configurable modules will save parachain teams time by eliminating the need to write their own implementations. Additionally, using standard components is a less error-prone approach, as implementation mistakes are more likely to occur when everyone is writing their own implementation.

We also plan to introduce a native token for the RegionX parachain, which will provide additional benefits. However, it is important to highlight that all functionalities related to the Coretime market, including fee payments, can be executed using the DOT token. This implies that interaction with the market will not necessitate ownership of the native token.

The governance of the chain will be **delegated** to **DOT holders**.

The development of the first phase will be divided into three milestones, by the end of which all the necessary components needed to deploy the parachain will be complete.

4.3. Milestones

Milestone 1.		
Task	**Description**	**Reporting & Delivery Methods**

Coretime XCM transfers design & research	Research and design the solution for reserve transferring Coretime regions from the Coretime chain to the RegionX parachain	The design is utilized in the development of the XCM configuration.
Coretime market componentes architecture design	Design of the architecture for the Coretime marketplace and its crowdfunding order module.	The design is utilized in the development of the Coretime marketplace and the order module.
Parachain configuration setup	Setup DOT as fee payment currency. Governance & treasury system setup.	The development of this task can be monitored on the public GitHub repository
XCM configuration & Custom AssetTransactor	Register the Coretime chain as a reserve chain for Coretime regions. Develop an AssetTransactor and other necessary components for handling the transfer of Coretime regions to the parachain.	The development of this task can be monitored on the public GitHub repository.
Regions pallet & ISMP integration	A FRAME pallet that will keep track of the cross-chain transferred regions. It will be integrated with the ISMP module to read Region metadata through state proofs from the Coretime chain. The pallet will be accessed from the XCM AssetTransactor.	The development of this task can be monitored on the public GitHub repository.
Frontend ISMP request handling	We will integrate the hyperclient npm package to handle the ISMP GET requests emitted from the RegionX parachain. This will be integrated into the RegionX frontend, allowing users to self-relay information and thus eliminating the need for a fee mechanism.	The development of this task can be monitored on the public GitHub repository.
Testing	The pallets will undergo thorough testing with unit tests. Additionally, we will write E2E tests to ensure the cross-chain region transfer functionality and the reading of region metadata through state proofs operate correctly.	The development of this task can be monitored on the public GitHub repository

Milestone 2.		
Task	**Description**	**Reporting & Delivery Methods**
Market Pallet	FRAME pallet that encapsulates the marketplace logic, migrating from the ink! contracts we previously developed. The primary modifications from the contract include: integration with the region pallet, support for both native currency and the DOT token for Coretime procurement, implementation of a fixed fee mechanism, and enabling access through XCM.	The development of this task can be monitored on the public GitHub repository
Orders Pallet	FRAME pallet responsible for handling incoming orders and the crowdfunding logic. This pallet will incorporate randomness to determine the end of the order period, making it difficult to snipe orders. Access through XCM will be allowed enabled.	The development of this task can be monitored on the public GitHub repository
Order creation UI	The RegionX frontend app will be enhanced to enable the creation of orders through the UI. This will be beneficial for new projects that do not have an existing parachain on Polkadot.	The development of this task can be monitored on the public GitHub repository.
Order dashboard UI	UI for browsing current orders on the RegionX parachain.	The development of this task can be monitored on the public GitHub repository.
Crowdfunding UI	UI that enables users to support market orders with their tokens.	The development of this task can be monitored on the public GitHub repository.
Testing	The pallets will be comprehensively covered with unit tests. Furthermore, we will write E2E tests where applicable	The development of this task can be monitored on the public GitHub repository.

Milestone 3.		
Task	**Description**	**Reporting & Delivery Methods**
Order processor design & research	Research aimed at designing the procedure based on which orders will be matched with the regions listed for sale. The goal is to ensure that orders with the most support secure the best deals on the market.	The design will be utilized in the order processor module.
Order processor pallet	FRAME pallet that is responsible for processing orders at the end of the order period. It will be loosely coupled with pallet-market and pallet-orders. This module will encompass the logic for matching orders with the regions listed on the market. Processing an order will involve making an XCM Transact call to the Coretime chain to assign the region to a specific parachain.	The development of this task can be monitored on the public GitHub repository
Coretime market & order processing documentation	It is essential to provide clear and easily understandable documentation of how the market and order processing operate, as teams interacting with them need to have a solid understanding of how they work.	The documentation will be published on the RegionX wiki.
Documentation on market integration	We will provide documentation for parachain teams planning to integrate with the market through XCM.	The documentation will be published on the RegionX wiki.
UI for tracking order processing	UI enabling users to track if an order will be matched with a region and, if so, with which region and at what price.	The development of this task can be monitored on the public GitHub repository
Testing	The order-processing pallet will be comprehensively covered with unit tests. We will also write E2E tests to ensure the entire procedure of purchasing Coretime from the market functions correctly: - A parachain makes an order - Order gets crowdfunding support - Region is listed on sale - The order period ends - The order is processed & the region is assigned to the parachain	The development of this task can be monitored on the public GitHub repository.

4.4. Timeline & Budget

Milestone 1 - Timeline			
Milestone	**Task**	**Time**	**Cost**
Milestone 1	Coretime XCM transfers design & research	50h	5000
	Coretime market componentes architecture design	40h	4000
	Parachain configuration setup	30h	3000
	XCM configuration & Custom AssetTransactor	70h	7000
	Regions pallet & ISMP integration	180h	18000
	Frontend ISMP request handling	30h	3000
	Testing	100h	10000
Milestone 1 Total		**500**	**50000**
Milestone 2	Market Pallet	180h	18000
	Orders Pallet	220h	22000
	Order creation UI	60h	6000
	Order dashboard UI	40h	4000
	Crowdfunding UI	60h	6000
	Testing	80h	8000
Milestone 2 Total		**640**	**64000**
Milestone 3	Order processor design & research	50h	5000
	Order processor pallet	220h	22000
	Coretime market & order processing documentation	40h	4000
	Documentation on market integration	40h	4000
	UI for tracking order processing	60h	6000

	E2E tests for the entire Coretime procurement process	100	10000
Milestone 3 Total		**510**	**51000**
Proposal Total		**1650**	**165000**

4.5. Timeline

Given that we have a team of three individuals committed to working on the project full-time, our objective is to complete all three milestones within 4 months of active development following the acceptance of the treasury proposal.

Considering we are pioneering the development on the novel Agile Coretime model and are the first team to create a Coretime marketplace, we will keep stakeholders informed about any issues related to timing. However, due to our thorough research, we're quite confident in our ability to deliver within the four-month period

Timeline	
Milestone	**Timing**
1	40 days
2	50 days
3	40 days

4.6. Payment condition

The development of the RegionX parachain is divided into three milestones, which would be funded by this treasury proposal. This proposal covers an estimated 1650 hours of work at an average hourly rate of $100, resulting in a total cost of $165000. This amount will be converted to DOT based on the EMA-30 price on the day the proposal is submitted.

Proponent: 12icMZXL8aMbKXC84XD4ASC4y22Zywi7ETgrksX7LZdjZwFA

Requested allocation: USD$165000 | The amount of DOT will be determined on the day of submission of the proposal based on the EMA-30 price(Subscan).

Governance referenda origin call: Medium Spender

4.7. Objectives/Success criteria

We will consider the project successfully delivered once new teams can get access to Coretime using RegionX at a price lower than what is offered in the Coretime Bulk sale. Another indicator of the project's success would be having existing parachains on Polkadot, which do not require full Polkadot Core, utilize RegionX to purchase smaller chunks of Coretime.

We would also like to see the development of Coretime trading bots, which will profit by purchasing Coretime in bulk, splitting it up, and then selling it for a profit.

Considering that cost savings and profit generation can be big motivators, we are optimistic about the high utilization of the RegionX market.

4.8. Communication strategy & Reporting

Throughout the project's development, we will report our progress through Medium articles, continuing the practice we have established (https://medium.com/@regionx).

In case of any questions, the team can be reached by sending an email to support@regionx.tech

5. Team

Sergej Sakač

Sergej is a member of the Polkadot Fellowship. He has been an external core contributor on Substrate and Polkadot for close to two years now. Sergej is also an Engineering alumni of the Polkadot Blockchain Academy (PBA) held in Berkeley.

Sergej Sakač Sr.

Sergej Sakac Sr. is an engineer with a master's degree and over 20 years of experience in organizational roles within the IT and engineering industries. At RegionX, he manages the project's operations.

Oliver Lim

Oliver is a Full-Stack blockchain developer who was involved in 3 projects granted by the Web3 Foundation. He worked with Sergej on RegionX from the start of the project.

Nathanael Liu

Nathan is a seasoned frontend developer with a rich background in blockchain development spanning over 5 years. Specializing in frontend development, he has left his mark on projects such as SushiSwap, TaoStats, DystopiaLabs, Based, and Bitcash.