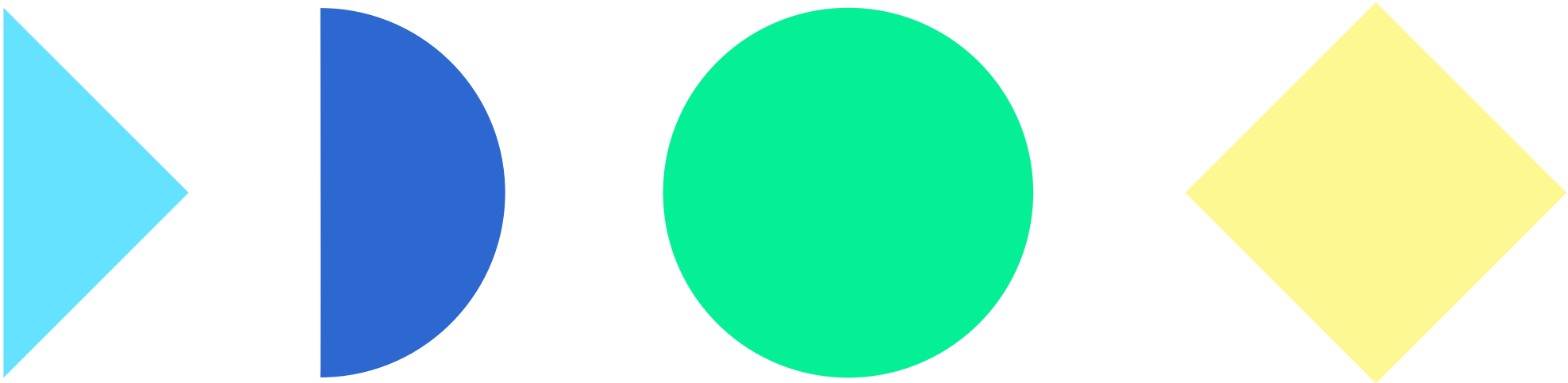
Ape Unit

Relevance Through **Developer Adoption**

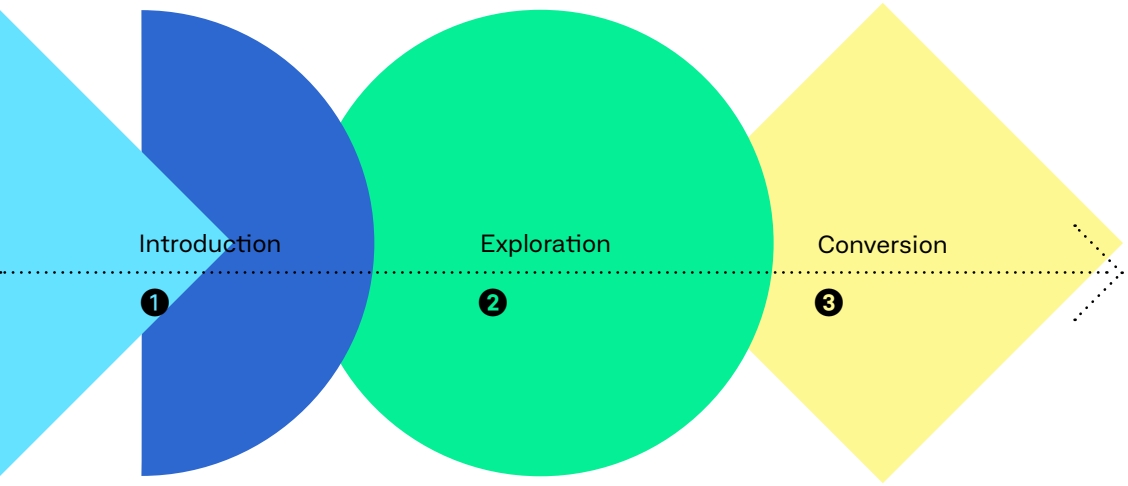
TABLE OF CONTENTS



Relevance Through **Developer Adoption**

Developers are the first fundamentally important target group of any game-changing technological project. Therefore one of the key factors of success is developer adoption.
Skilled and engaged developers massively increase the quality and stability of a new technology.

Adoption Journey



ABOUT US

Ape Unit is a network of designers, developers and strategists with a special unit focusing upon emerging technologies and DLT in general. We see many opportunities and potential use cases for emerging technologies. At the same time we know ground work has to be done before new technologies can reach their fullest potential.

To facilitate the adoption of emerging technologies, we focus upon the Developer Experience as a whole.

Our goal is to make a good project into a relevant one. What we call Developer Experience enables developers to interact with a technology platform and helps them to create something (ideally meaningful) on top of it. Being based in Berlin provides us with the opportunity to tap into a vibrant culture of curious developers.

Our team consists of senior developers who are experienced in managing large projects and have knowledge of different programming languages. They work closely together with our designers, since good ideas need intuitive interfaces and clear visualisations.

How we work on **Developer Experience**

Developer Experience (DX) is the equivalent to User Experience (UX), with the change that the target user is a developer. DX comprises of a 360 view on the engagement a developer has with your technology.

Developers nonetheless, like everyone else, would like to work in a welcoming environment that is stable, clear and easy to comprehend. They look for stable environments with supportive SDKs, tooling and clear documentation.

WHY DEVELOPER EXPERIENCE MATTERS

Big software development projects often are a challenge to coordinate and have a steap learning curve for external developers. This is due to their complex nature with many moving parts. Just hiring more engineers, will not make the development process more efficient, as it only will create more overhead. Hence Brook's law: "adding manpower to a late software project makes it later".

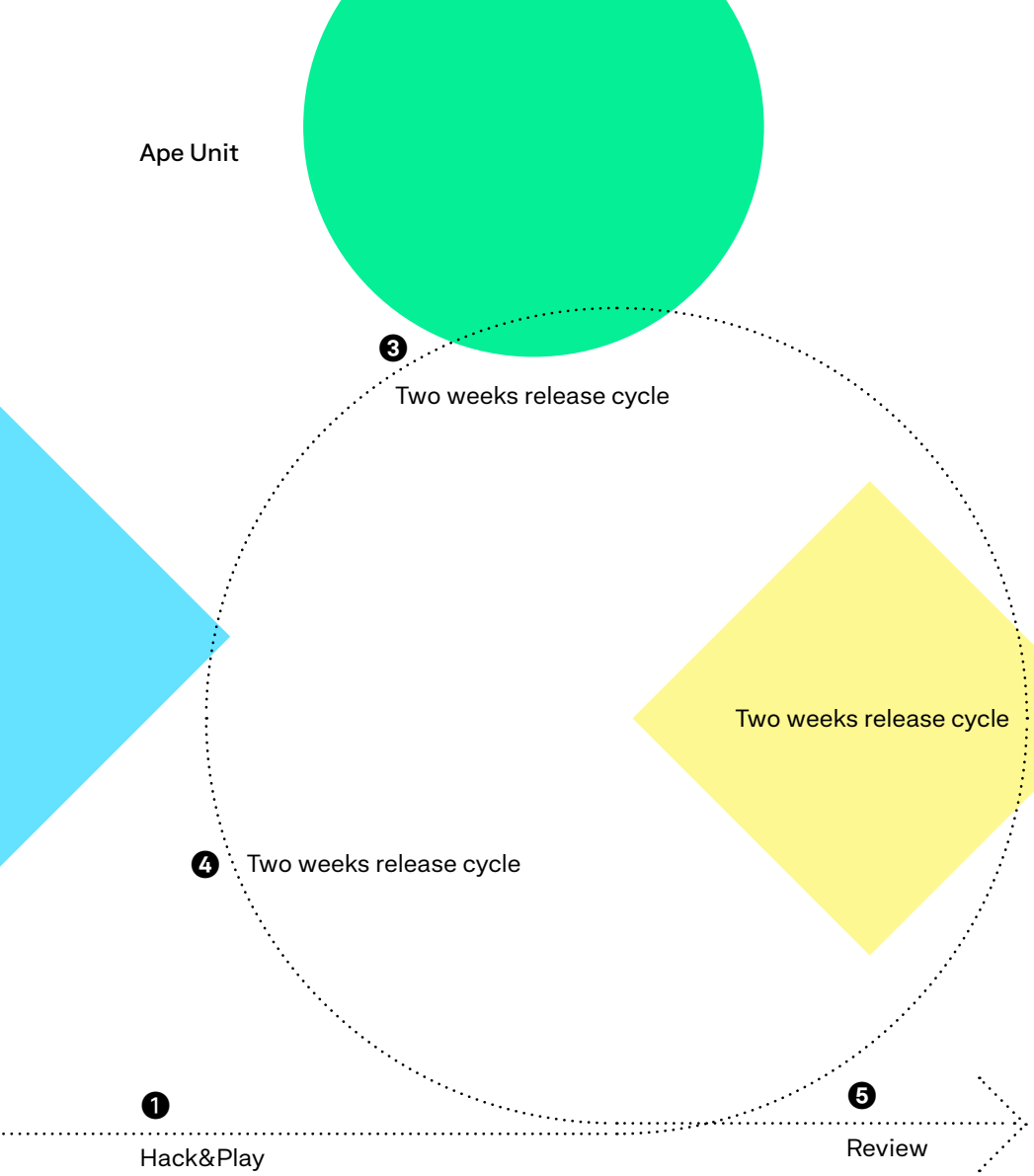
This is why it is important to focus upon developer experience.

Commonly, developer tools and environments are mainly reviewed for technical correctness, but can be difficult to use and not well documented.

Often projects fall short on this, as coordinating, planning and documenting the release of components across multiple teams, is not an easy task and requires a user-centric focus.

Having developers wanting to build upon your technology is key for adoption. Therefore a good DX is important to the success of your project

Our team builds upon years of of experience to improve the accessiblity of new technologies, create an intuitive developer experience and develop new processes to tackle the challenges that come with paradigm shifts.



1

DISCOVERY PHASE HACK&PLAY

We will start with a short initial assessment, to get a clear overview of what are the actual needs and will be the scope of the project
For this, we created Hack&Play, a developer workshop that has as a goal to spot weak and strong points of a techstack. During the workshop external developers will sit together with our team for one or two days and write code examples and develop small prototypes.
Our team documents how people are using the tech-stack, identifes blockers and spots potential improvements.
This will result, next to the developed prototypes and code examples, in a comprehensive overview of the strengths and weaknesses of the existing code and documentation.
The report will give clear directions for improvement and insights into how the developer community experiences the tech-stack.

2-4

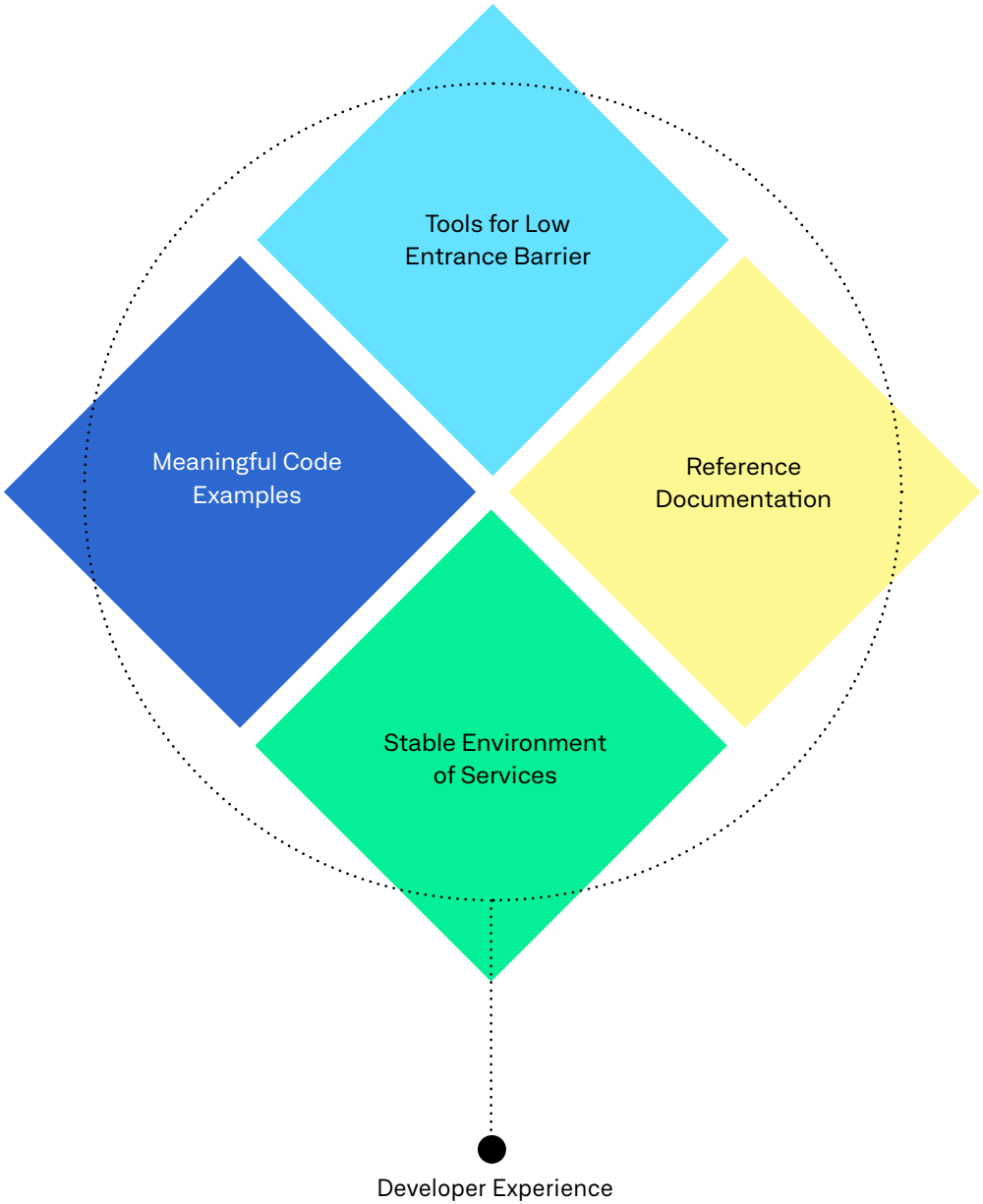
IMPROVEMENT OF ONBOARDING EXPERIENCE

After the Hack&Play we will set specific goals and success measures for improvement. A dedicated team will be assembled in order to improve the existing onboarding material and existing documentation.
Our emphasis is on producing quality code and documentation which fits its purpose. We achieve this through strict code review policies and test coverage requirements.
In order to connect the tech with the real world, we we will build use case specific code-examples, tutorials, prototypes and optimise the SDKs accordingly. We prefer to work with specific use-cases and/or partners that would like to integrate the technology in order for the tech to be tested and demonstrate real world value.
We usually follow a two weeks release cycle where we communicate and coordinate our work to other teams and communities and follow practices like semantic versioning and meaningful changelogs to keep everybody informed.

5

REVIEW

After 4-6 weeks, we organize a second Hack&Play session where we build a diverse set of prototypes in order to stress test our adjustments and review progress. This gives room for evaluation and readjustment of goals if needed.



Typical examples are

1. **Bridge technologies, like middlewares**
2. **SDKs**
3. **Tooling (CLIs, Plugins for common editors, WebIDEs)**
4. **Specification of common standards to coordinate development across different teams and projects**
5. **Prototypes to demonstrate specific features**

4

FURTHER DEVELOPMENT

Based upon the review, we will optimise the direction of the developer experience. Being familiar with the technology at this point will give us the opportunity to redefine the needs and services delivered. We can continue with a sole focus upon the onboarding experience, or dive deeper into the development of specific building blocks.

One of the biggest challenges for evolving ecosystems is the coordination of different development teams pursuing their goals and agendas. Configuration management of complex projects with many moving parts is one of our specialities. This includes developing, coordinating and planning the release of components across multiple teams, communicating with customers and the community, and ensuring there is always a stable environment of services and libraries deployed.

How we work on **Developer Engagement**

Even the best technology will not prosper if no one makes use of it. Many technological projects have onboarding tutorials and documentation available online, but fail to attract and retain the interest of developers. An additional problem is that digital engagement is hard to measure beyond clicks and views.

To tackle these problems, we have developed a p2p learning platform called dacade, on which developers can get to know a certain technology and are incentivised to actively engage.

ABOUT DACADE

On **dacade.org**, developers are rewarded to follow tutorials, test and build upon the technology and leave reviews to help their peers. Dacade facilitates the active monitoring of development effort, not only giving quantative results such as the proportion of users who finish a course, but as well qualitative insights as the code being used or use cases being programmed.

BY THE NUMBERS

4K

LEARNERS

Have already registered on dacade.

1K

SUBMISSIONS

To the challenges by Learners who finished the community tutorials.

3K

FEEDBACK

Learners gave to submissions of their peers.

Ape Unit

P2P Learning Deliverables

1. Incentives for active social interactions to push adoption
2. Programming challenges to showcase the technology
3. Measurable results
4. A dedicated community of developers

PEER-LEARNING AT DACADE

One of the biggest issues in active learning environments is offering individual feedback in a sustainable manner. Access to sufficient teaching resource is an expensive bottleneck. Dacade solves this by enabling a p2p learning environment.

On dacade, developers can make use of curated learning material, solve creative programming challenges, apply their new knowledge practically, and give and receive peer-feedback. The p2p educational interactions can be incentivised with rewards to encourage engagement. The submissions to the programming challenges are often projects on GitHub, which will result in an accessible body of practical examples.

DISCOVERY

There is a great deal of competition for the attention of developers. Traditional marketing efforts such as sponsoring a conference, are often expensive and hard to measure. Hackathons and programming challenges can easily be set up at dacade, and will be more easy to monitor, cost-efficient, scaleable and bring meaningful results.

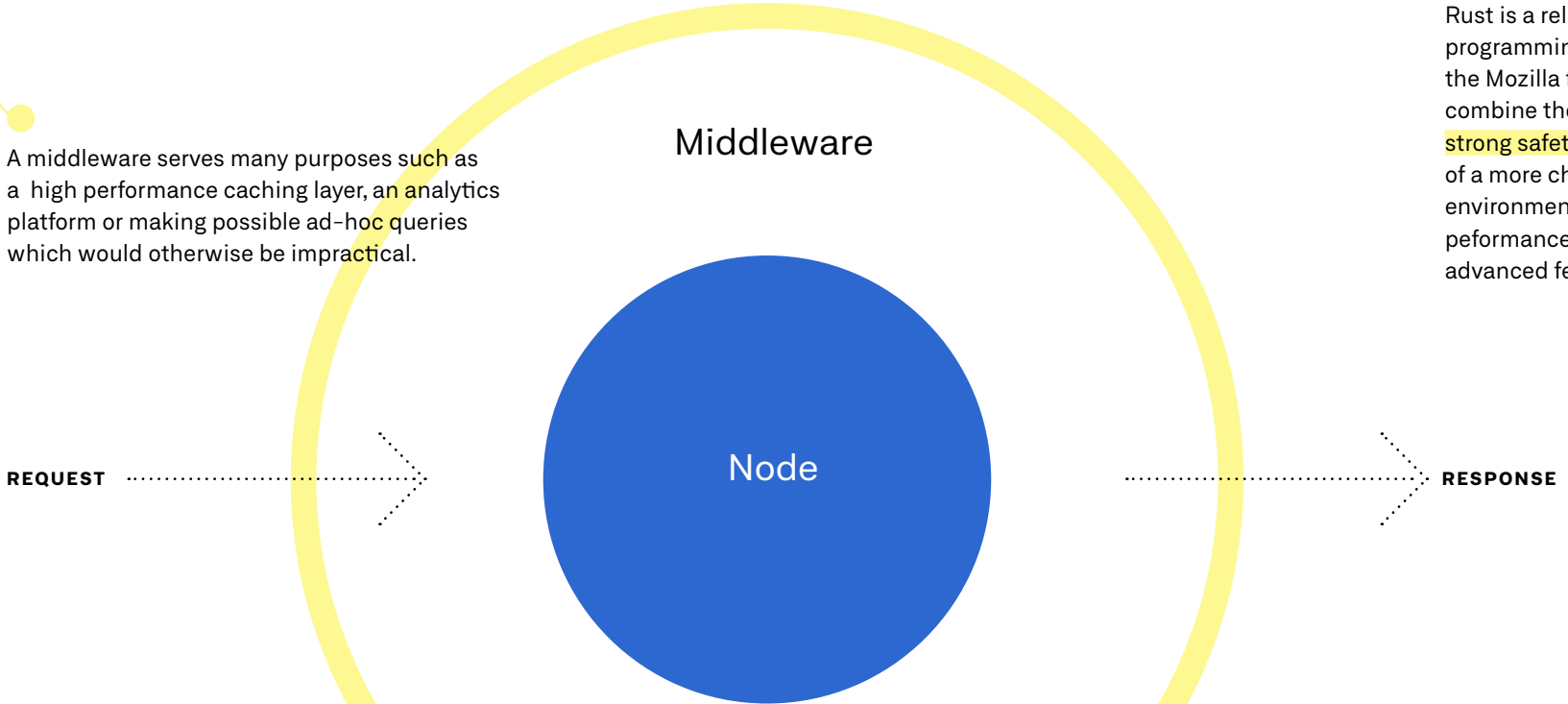
LEARNING ENVIRONMENTS

The learning environments are designed with a strong focus on gamification aspects and attractive rewards structures in order to create an active online developer community around a technology. The platform is furthermore an excellent tool to onboard new developers to a technology during an event or live workshop.

Case Study
Middleware

The purpose of a middleware layer is specific to each blockchain for which it is designed. In general, the role of a middleware is to correct for the general lack of interactive performance and reporting in a blockchain. It is an open source software which can also be run as a server (SaaS). Having a custom middleware helps the development of blockchain use cases, as most substantial blockchain projects will ultimately require more services than a node on its own can provide

A middleware serves many purposes such as a high performance caching layer, an analytics platform or making possible ad-hoc queries which would otherwise be impractical.



DETAILS

Due to the conflicting demands of decentralization and trustlessness, on the one hand, and performance and functionality on the other, our philosophy requires that our solution is transparent and auditable. What this means in practice is that the source code is open, that users may run their own instances, that we regularly provide our own database dumps, and that the software is able to audit itself.

The basis of the middleware is a server process, written in the Rust programming language, and a storage layer, using the PostgreSQL relational database server. Rust is a relatively new systems-level programming language, originally from the Mozilla foundation, which aims to combine the performance of C/C++ with strong safety guarantees, at the expense of a more challenging programming environment. PostgreSQL is a high-peformance SQL database, with many advanced features.

Possible use cases

The middleware layer can support a large number of use cases, we list a few below

- **Business analytics**
- **Fraud detection**
- **Real–time data visualisation**
- **Providing a high–level API for applications**

Our goal is to model the blockchain as generically as possible, using, for instance, JSON searches where we can, in order to be able to reuse as much code as possible between blockchains. Where we must, we unpack data such as contract calls and results, and build auxiliary tables for them. On top of this we provide a rich set of REST calls, consuming and producing JSON. We build this in front of a blockchain node, in order that the middleware enhances the nodes functionality and presents it unchanged as far as possible. Where a node call is slow, we may proxy it invisibly, returning data from our own database.

SCHEDULE

1. The first step, as previously described, is a research exercise, in order to ascertain which functionality the node lacks, and where it is too slow. In this regard, development or examination of a blockchain explorer is instructive–a blockchain explorer is the classic interactive blockchain client, and exposes many limitations.

2. Next, we build a simple middleware which seeks only to correct the issues identified above and to keep itself up to date with the state of the underlying blockchain. This phase of development would be to agree on the specific functionality, which may include several of the following.

   - **Caching blocks for faster explorer performance**
   - **Decoding smart contract calls and return values**
   - **Storage of account information**
   - **Calculation of block rewards**
   - **Metrics on blockchain performance**

3. From this point on development is driven by the needs of users. We found from our experience with the æternity blockchain that as soon as users have access to the middleware they tend to be quite forward in expressing their needs.

If you have been a part of our journey
so far, we want to thank you.
If we haven't met yet, we'd love
to hear from you.

Emil Wagner
Managing Director

emil@apeunit.com