Date of acceptance Grade

Instructor

APM Requirements Analysis and Comparison for Veikkaus Oy

Topi Törnroos

Helsinki February 10, 2021

Master's Thesis
UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta/Osasto – Fakultet/Sektion –Faculty/Section	Koulutusohjelma – Studieprogram – Study Programme
Faculty of Science	Department of Computer Science

Tekijä – Författare – Author

Topi Törnroos

Työn nimi – Arbetets titel – Title

APM Requirements Analysis and Comparison for Veikkaus Oy

Ohjaajat – Handledare – Supervisors

Jukka Nurminen, Tuomas Lempiäinen

Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages
Master's Thesis	10.02.2021	37 pages + 6 appendix pages

Tiivistelmä – Referat – Abstract

Application Performance Management (APM) is a growing field, and APM tools on the market tend to be complex enterprise solutions with features ranging from traffic analysis and error reporting to real-user monitoring and business transaction management.

This thesis is a study done on behalf of Veikkaus Oy, a Finnish government-owned game company and betting agency. It serves as a look into the current state-of-the-art field of leading APM tools as well as a requirements analysis done from the perspective of the company's IT personnel. A list of requirements was gathered and scored based on perceived importance, and four APM tools on the market—Datadog APM, Dynatrace, New Relic and AppDynamics—were each compared to each other and scored based on the gathered requirements. In addition, open-source alternatives were considered and investigated.

Our results suggest that the leading APM vendors have products very similar to each other with marginal differences between them, feature-wise. In general, APMs were deemed useful and valuable to the company, able to assist in the work of a wide variety of IT personnel, as well as able to replace many tools currently in use by Veikkaus Oy and simplify their application ecosystem.

ACM Classification:

- Networks~Network Performance Evaluation
- Networks~Network Services~Network Monitoring
- General and Reference~Cross-Computing Tools and Techniques~Performance

Avainsanat – Nyckelord – Keywords

APM, Application Performance Management, Application Performance Monitoring, Requirements Analysis

Säilytyspaikka – Förvaringställe – Where deposited

Muita tietoja – Övriga uppgifter – Additional information

Contents

1 Introduction

1.1 Veikkaus Oy

The 2017 merger between the three companies of Veikkaus, Finland's Slot Machine Association (RAY) and Fintoto together into a single company, Veikkaus Oy, necessitated an infrastructure change. As each company had its own servers, their own codebases and their own customer data, a new Veikkaus website and a mobile site had to be designed to accommodate for all the previous customers and all the services offered for each of the three companies.

Even after the merger, however, the online presence of Veikkaus was split across a multitude of physical data centres, such as ones owned by Fujitsu and Elisa, and the company's own servers, majority of them physically located in Pähkinärinne, Vantaa.

It was clear a solution of some sort would have to be found, but it was not until 2019 that it was conclusively decided that the company would seek a cloud-based solution, at least for their online gaming infrastructure. The arguments in favour were uniformity, scalability and the fact that the current development in general happens to favour cloud services, and newer technologies tend to be developed with them in mind. According to current timeframe, the migration of majority of gaming services into the cloud will be finished by the end of 2022 at the very latest.

The need to revamp the network infrastructure has also presented an opportunity to improve upon it. The Veikkaus infrastructure is rather complex, consisting of many possible error points and various services—not all of them maintained by Veikkaus—that may fail. As such, there have been calls for something to help with regarding error catching and performance monitoring, among other things. The consensus within the company has been that APM might offer the solution, or at least be a major part of it.

1.2 APM

The term "APM" refers to either "Application Performance Management" or "Application Performance Monitoring". Many sources use both terms interchangeably, though some manufacturers, such as AppDynamics, prefer drawing a distinction between them [Adr19]. For the purposes of this paper, though, APM will henceforth refer to Application

Performance Management specifically, as defined below.

One of the main functionalities of an APM is monitoring the performance of large software applications, by continuously collecting data and measuring application performance, as well as helping to find and diagnose problems related to those metrics. Yet, despite what the word 'performance' may lead one to believe, APM is not simply a tool for detecting performance anomalies. In practice, the modern APM is an all-inclusive monitoring solution with a set of tools that extend far beyond workload related issues. With features ranging from code-level error detection to real-user monitoring and business transaction management, commercial APMs of today are fully featured enterprise applications that offer potential value in a multitude of ways, with features useful to a wide variety of potential interest groups.

Many tools calling themselves APMs exist on the market, ranging from both commercial to open-source applications. Taken as a whole, they form a very disparate set, with wildly varying features and implementations. As such, it is important to firstly define what an APM even is.

In their paper *Application Performance Management: State of the Art and Challenges for the Future*, Heger, van Hoorn, Mann and Okanović outline four distinct APM activities: data collection, data storage and processing, data interpretation and use, and data presentation [HHM17]. In other words, APM is not a mere passive collector of data: it is also a tool for translating it into understandable and useable metrics. According to them, the gathered data may be interpreted in a number of ways, three of which they lay out explicitly: problem detection and alerting, sending alerts in case of anomalies to be sent to system operators; problem diagnosis and root cause isolation, to actually pinpoint the nature of the anomaly and thus assist in correcting it; and system refactoring and adaptation, to identify bottlenecks and assist in improving performance in general.

Gartner Research, however, has their own three-dimensional APM model that defines an APM tool as one that fits the three following functions [FeC20]:

- *Front-end monitoring*, specifically concerned with the experience of real users of a web service.
- *Application discovery, tracing and diagnostics*, mapping the relationship of different components within the application and the transactions that occur within—which may sometimes contain an end-user experience aspect to it.

- *Analytics*, or automatically detecting causes of problems within the application and generating useable reports regarding them.

Gartner's definition is much stricter than the one used by Heger, van Hoorn, Mann and Okanović. It does not suffice for an APM to merely collect data, they must correct it through both front-end focused real user monitoring and back-end focused application diagnostics and relationship mapping. Mere traffic monitoring or log analytics aren't enough, which excludes many commonly used tools—such as Splunk—from being considered APMs to begin with.

Gartner's stricter model will be used as a basis for the definition of an APM in this paper. The above three dimensions will be further elaborated later on in the paper, particularly from the perspective of the needs of Veikkaus.

Veikkaus had an earlier APM project back in 2017–2018, where some requirements were drawn, and some proof-of-concept demonstrations held. The general feeling left on the Veikkaus employees who had worked on the matter was a positive one, with many of those involved with the project reporting having been impressed with the tools and feeling they would bring in value for the company. The actual acquisition, however, never ended up happening due to a lack of offers from the APM providers, after which the APM project was merged into a larger attempt to renew Veikkaus's infrastructure. This, too, ended up being cancelled due to reasons unrelated to APM itself.

The actual need for an APM never disappeared, however, so a second proper project to acquire an APM tool was begun in late 2019—this time with more research and better preparedness from Veikkaus's side.

1.3 Research Questions

As there are multiple distinct interest groups within Veikkaus Oy that all have their own wishes regarding the selection of the tool, picking the right tool is not a trivial task. Different APM tools have different features and focuses, and unless the requirements of Veikkaus are elicited and prioritised properly, choosing "the best" from among them is an impossibility. As such, our first research question is as follows:

- RQ1: What are the various requirements regarding APM that different interest groups with Veikkaus have?

- What requirements could the tool satisfy?
- What potential problems with the tool do they want mitigated?
- Where are their priorities? What requirements are considered more important than the others?

Ultimately though, APM is only a means to an end. One should not narrow their vision squarely on APM alone, especially if there are less costly and less difficult to implement alternatives:

- RQ2: Are any of the above requirements solved easier by some other means?
  - Are some of the requirements gathered not something that one should rely on APM for to begin with?
  - Should some of the requirements gathered be considered lower priority due to there being alternative methods to solve the problem?

Of course, selecting, analysing and filtering the requirements is only the first phase. The requirements will finally help us select the proper tool:

- RQ3: Which APM tool is the best fit for Veikkaus?
  - Which tool best fits the requirements specified in RQ1?
  - Are there any lower cost alternatives that are not a significant downgrade compared to the one chosen as the 'best one'?

Ultimately, the goal of this paper is to assist Veikkaus in selecting the right tool for its own needs, either directly or to serve as a basis for competitive tendering. At the same time, it should provide insight into requirements specifications within a major organisation, as well as a broad look into the state of the art of the APM tools on the market.

1.4 Methodology

As the primary purpose of the paper is to identify the specific needs that various teams within Veikkaus have, and to see if an APM could help to alleviate the current problems being faced, the primary means of gathering data was through interviews and requirements elicitation with Veikkaus employees. The process was as follows:

1. The key interest groups within Veikkaus were identified: the units and teams most likely to benefit from APM and to have an interest in it.

2. A first open interview was conducted with a representative of each interest group, to get a feel for the groups' unique perspectives and needs. A set of preliminary requirements were created based on the preliminary interviews, knowledge of APM's features and previous experiences.
3. Larger interviews were conducted with each group, with multiple representative present, where everyone had a chance to pitch their own ideas and requirements.
4. Key representatives from each group were singled out to separate the gathered requirements into mandatory and optional ones, while giving the optional requirements a weighed score based on their group's viewpoint.
5. In a fourth and final interview, representatives of each interest group were gathered together to collectively discuss each requirements point by point, while assigning them all a final score based on mutual consensus.

The elicitation process was sufficient for getting a clear view of the needs, expectations and worries regarding the APM. Following the elicitation, a study was performed to see which of the requirements were satisfied by which products on the market, commercial and open source alike. This was performed both by reading the respective documentation, and by directly questioning representatives of the companies in charge of the products.

2 Utility of APM to Veikkaus

2.1 The Infrastructure

Though the merger of the three companies introduced its own complications to Veikkaus's network infrastructure, the systems in all three companies were complex enough as it was. The company's own systems communicate with banking services, the systems of Digital and Population Data Services Agency (Digi- ja väestötietovirasto) as well as the systems of third-party online gambling vendors that Veikkaus cooperates with to offer games and sports betting services; the laws and regulations on both national and EU level likewise require attention, with additional services constructed to ensure they are complied with. A tangible example of this is, for example, *Pesukarhu*: a service integrated into the rest of the online casino to monitor and alert about potential money laundering suspicions.

In addition to a plethora of distinct services that must be in communication with each other, their actual physical distribution makes for a challenge as well. As of right now, the

online casino services reside partially in Amazon's AWS and Google's GCP cloud services, with some servers managed by Elisa and some by Telia, in addition to some physically residing in Veikkaus's own Pähkinärinne facilities. With how security critical everything is, rigid discipline must be maintained when it comes to deciding what parts can be allowed to communicate with others—but of course, some communication must take place and some openings must exist for it to happen.

A single request may go through a multitude of services and may pass through AWS, GCP and the physical data centre, all. Following it through its whole path has been difficult, even nigh impossible; this makes finding errors frustrating and slow, and if it gets blocked by some security rule or a firewall somewhere, finding the culprit has often required some trial and error.

2.2 Interest Groups within Veikkaus

For the purposes of this paper, six distinct interest groups have been identified within the company. From the online casino development side, it is important to take the needs of both *front-end developers* and *back-end developers* into account. The former are represented by the Veikkaus.fi front-end development team, and the latter by the 'Kundi' team, responsible for developing the customer services system for Veikkaus.

Veikkaus has a large Quality Assurance unit that could see utility from APM, and in this work, they are represented by the *performance testing* team, who stand to be direct beneficiaries of the performance measurement related aspects of the tool.

The *Infra & Security Services* unit is represented by two distinct groups: from the perspective of the general *infrastructure* team who holistically maintain the company's network presence, as well as the *networks* team who are responsible for performing the firewall openings and ensuring communication between the systems and services remains functional and secure.

Finally, there is the *production support* team, responsible for identifying and potentially fixing issues that occur in production. Should the login service go down, for example, it is their task to discover both the problem and the root cause of it.

These six groups, *front-end development, back-end development, performance testing, infrastructure, networks* and *production support*, are by no means the only valid groups within Veikkaus to focus on. There are some 250 employees working in the IT side of

things from Veikkaus alone, and that number does not even include the numerous consultants from other companies. The QA unit alone is nearly 50 employees strong and could be represented by more than just the performance testing team. It is also plausible there could be interest in the APM tool outside of the IT function of the company as well: marketing, for example, could be interested in the end-user experience statistics that many of the products gather.

Yet the choices were not arbitrary. The six groups identified above are all vitally important for the functioning of the company's online casino services, and they all have identifiable problems that the APM could potentially solve. Amongst each of the six groups a representative could be found with experience with, or interest in, APM. And though all the groups work within IT, they each had distinct needs and perspectives into what the product could and should do—and what it should not.

To properly classify and categorise the distinct needs of the chosen interest groups, we will use Gartner Research's three-dimensional model as the basis for our analysis.

2.3 Three-Dimensional APM Model from Veikkaus's Perspective

2.3.1 Front-end monitoring

Gartner describes front-end monitoring as a subset of a broader concept of Digital Experience Monitoring (DEM), defined by their words as *"a performance analysis discipline that supports the optimization of the operational experience and behavior of a digital agent, human or machine"* [GSR19]. Though many APM tools incorporate the full spectrum of DEM functionality, only the front-end monitoring aspect is considered to be strictly in the realm of APM. Gartner further clarifies that the definition of front-end monitoring includes real-user monitoring (monitoring the transactions performed by the actual end-users of the application the APM is deployed for) and synthetic monitoring (using automation to simulate actions the application is expected to be able to perform).

As such, front-end monitoring comes with a passive and an active component. Passively, the APM follows the actions performed by real users of a web service, monitoring their behaviour and transactions and automatically alerting the APM's users of any problems that might arise in real time. Actively, it is able to use scripts to simulate user behaviour during times of less use, or to check on services that are not as actively accessed by ordinary users.

From the perception of Veikkaus, front-end monitoring comes with one clear benefit. From the perspective of production support in particular, the active monitoring and real-time alerts that many APM tools provide will aid them in responding to potential issues that arise much faster than now. So far, the production support team has often been reliant on an end user of Veikkaus.fi contacting the customer support through the website's chat or helpline when something goes wrong in production—when they are, for example, unable access specific games due to a critical service being down. The real-user monitoring (RUM) aspect would alert the team to issues much faster, potentially reducing the amount of time from the problem occurring to it being discovered and corrected.

Many APM tools focus on not merely watching for any issues with the web service itself, but also on the behaviour of customers in general. APM tools provide concrete metrics such as API roundtrip times or First Meaningful Paint times, which can be valuable for determining whether a change to the code has been a positive or a negative one from an end-user perspective, or whether there are any compatibility issues with specific end-user devices or browser types [Bal19]; the deeper analysis that APM offers can further refine those metrics into more tangible data, with some APMs having functionality for translating metrics such as page load times into tangible business value [Blo18].

The metrics gathering aspect of front-end monitoring is particularly prized by the front-end development team: while they already have tools capable of gathering such metrics (such as SpeedCurve), they tend to be simpler and their measurements less in-depth: they may be able to measure front-end load times, but APMs offer full-stack visibility. They are able to follow a particular end user on their path through the application, from the time of first load all the way to the end of their session, measuring every transaction they make and dissecting every individual request they make as it passes through the complex layer of services Veikkaus.fi uses. This holistic approach is valuable to the back-end developers as well: while they might be able to measure the performance of a particular service by, for example, sending a test request to their API, APM's way of measuring performance from the perspective of real end users in production environment in the actual context of things should at the very least provide another perspective on things.

As much of front-end monitoring revolves around performance measurement, it's only natural for the performance testing team to have an interest in this dimension as well.

Their perspective is rather different, though: rather than being interested in the Veikkaus.fi production side of things, the performance testing team has their own testing environment they are using. Without any real end-users, the real-user monitoring aspect of the APM is less useful for them; however, the synthetic monitoring aspect might be quite helpful in comparison, allowing them to create scripts to simulate transactions for the tool to measure and analyse.

2.3.2 Application discovery, tracing and diagnostics

Application discovery, tracing and diagnostics, or ADTD, is defined by Gartner as "a set of processes designed to understand the relationships between application servers, to map transactions across these nodes, and to enable the deep inspection of methods and other host resources.", and further splits ADTD into three distinct categories: application topology discovery and visualization, user-defined transaction profiling and application component deep-dive.

Topology discovery is one of the primary features of a modern APM tool. As Appdynamics puts it in their blog, an APM tool should be able to automatically discover the relationships between application components in how they communicate and display them as a graphical relationship [Hir19].

From the perspective of Veikkaus's infrastructure unit and network team, this feature is crucial. Veikkaus has fully embraced the modern standard of distributed, componentised microservice applications. As powerful as such an approach is, it does lead to a dramatic increase in individual components in Veikkaus's own systems, and the increase in components means an increase in potential points of failure.

Due to how complex the Veikkaus.fi online gaming infrastructure is, it can be difficult to understand the relationships between its various components; furthermore, when it comes to making firewall openings or diagnosing why a given request fails to get through, it would be helpful to be able to visualise the full path a specific request takes and to see, for example, which physical or virtual device it fails to pass through. As of now, there are no tools in use that can replicate this functionality, and though hand-drawn graphs of Veikkaus.fi's network infrastructure exist, they only go so far in helping to trace down related problems.

What Gartner and Appdynamics call user-defined transaction profiling is also known as

business transaction monitoring [Dyn20], but both refer to the same essential concept: the ability to define specific user actions as concrete “business transactions” or BTs, tracking and measuring them across the various components the request or requests pass through [Dyn20]. From the perspective of Veikkaus, a useful business transaction might be, for example, one that tracks a user as they arrive on a sports betting page all the way down to the moment the bet is placed and goes through. The ability to comprehend specific user interactions within Veikkaus.fi as distinct units should be of use in monitoring the health of the system as a whole, allowing the production support team to be alerted if issues arise within them, or for development and infrastructure to see how changes made in the system are reflected in real use cases by the actual end users.

The final aspect of ADTD—application component deep-dive—is closely related to the above two concepts. As AppDynamics defines it, it refers to recording the internal workings of the application components [Hir19]. For example, recording the call stacks or applications or individual SQL queries made to database servers, storing exceptions and statistics. As such, for example, if the end user’s bet fails to go through due to an API error, the APM is capable of showing the exact query and the exact error code returned by the API, pinpointing the problem rather than merely showing where it happened.

All in all, ADTD is concerned with troubleshooting. The three aspects above combine together to form a map of the application systems and services, to show the paths of specific end-user transactions made, and homing in on problems where they happen, displaying the exact component and even the exact lines of code in which they occurred. It is not only useful for catching explicit errors, but also finding performance issues and similar misbehaviour. A critical element of APM, each of the six interest groups reported finding useful aspects in it as a whole that would benefit their work.
In their book APM in the Digital Enterprise: Managing Applications for Cloud, Mobile, IoT and eBusiness, Sturm, Pollard and Craig point out the value of APM when it comes to comprehending complex systems and the transactions within, specifically singling out topology modelling and transaction tracing as useful features [CPS17]. According to them, component-based applications—such as Veikkaus’s gaming services—require proper visibility to end-to-end execution, something that Veikkaus is currently lacking in.

2.3.3 Analytics

The final dimension of APM is concerned with processing the data gathered by the tool

and generating reports and alerts based on what is found. Anomaly detection is, after all, of little use unless those anomalies are accurately reported to the end user of APM. By Gartner's definition, *"The APM solution must provide domain-centric artificial intelligence for IT operations (AIOps) functions, using AI/machine learning (ML)"*. In practice, this means the APM must more or less be independently able to detect and report anomalies without the end-user of the tool having to define exactly what counts as one. In the context of Veikkaus, this could be something as simple as automatically identifying a misconfigured firewall or a crashed Docker container somewhere in production, or something more complex such as detecting an unusually high number of failed logins and using it as a basis to report a potential problem within login services.

Different interest groups within Veikkaus have different needs regarding analytics. From the perspective of production support, for example, it is enough for them to know that the problem exists and where it lies so they can forward the issue to the correct team to manage—depending on whether the issue stems from a malfunctioning container or an error within backend code, for example, the responsibility for correcting it falls to a different team within the company. As of now, there can be a lot of juggling back and forth to try to find the root cause of production issues. Solving this issue is not just a matter of convenience, it is quite crucial: Amazon for instance estimated in 2013 that their service's downtime cost their company $66,240 per minute [Cla13], and though Veikkaus is a much smaller company, there is no doubt that a swifter response to incidents might result in a significantly reduced loss of potential revenue.

Sufficiently advanced analytics should be of some use to the developers as well, with an ability to perhaps notice problems in the codebase or configurations that are not otherwise obvious. The performance testing team might be interested to get useable reports of genuine performance problems within production environment, both to aid in discovering them and in configuring the performance testing environment to be better in line with production.

2.4 APM Feature Breakdown

While the three-dimensional APM model is sufficient for discussing APMs in the broad terms and for determining what or what does not count as an APM for the purposes of this paper, a more granular approach is needed to get useable insights for Veikkaus in particular.

Much of an APM's value comes from how interlinked their various features are, and drawing distinctions between some of them may be somewhat artificial. Yet, an APM is a complex whole, and over the course of a study, it has become apparent that the best way to explain an APM to someone not already familiar with it is by focusing on individual things it can do to help in their work.

The following breakdown is essentially arbitrary. It is not meant to be an all-encompassing whole to describe every single individual feature an APM has; neither is it meant to represent shared, universal features that all APMs have. Instead, it's a list of features that many APMs currently on the market have that are, based on the interviews performed, relevant to the needs of Veikkaus at this time.

2.4.1 Instrumentation and Full-Stack Tracing

The core principle behind an APM's functionality is that its ability to *instrument* code, typically by having it automatically injecting its own *monitoring agent* across the full stack of the service being monitored. The front-end code is usually injected with a JavaScript agent, and similar agents are deployed for back-end services as well. Databases, physical and virtual hosts, Docker and Kubernetes containers, AWS Lambdas and GCP Cloud Functions are all monitored either with the agent or with a cloud platform's own tools. Generally speaking, no manual changes to the actual code as written must be performed, though some extra analytics features may require this.

Modern APMs tend to support all commonly used technologies: some relevant to Veikkaus are Java, Python, NodeJS, .NET and a variety of database frameworks, all of which are supported by the APMs reviewed in this paper.

As every component of the service is instrumented, every call is fully traced across the whole stack. Should a login-related API call be made, for instance, the APM track the full impact the call has on the whole system: it is able to trace not only the initial request to the login service itself, but also follow through on any subsequent database requests or further calls to other services made as a result of the initial request.

The data gathered in this manner is used as a basis for much of APM's analytics capabilities. Under ordinary circumstances, it may be difficult and time-consuming to troubleshoot an issue with slow logins. An APM, in theory, would be able to tell that for instance that a particular verification service or a specific database server was the source of the

slowdown, and would ideally even able to point the specific function that was bottlenecking the call.

2.4.2 Real-User Monitoring

Real-User Monitoring, or RUM, is an integral aspect of the *Front-End Monitoring* dimension of Gartner's three-dimensional model. By typically instrumenting the front-end of the application with JavaScript agents, the actions of real end-users are tracked and followed. What sets an APM's RUM capabilities apart from those of common monitoring tools' (for instance, Google Analytics') is how it interacts with APM's full-stack monitoring capabilities. A user's every action is not merely recorded in a vacuum: rather, they are further traced through the whole stack, so that the APM can tell what effects a specific click of the login button had in the user database deep down in the line.

Depending on the APM and the settings used, different kinds of sampling rates may be used. With some solutions, tracking and recording 100% of user sessions is a possibility, but with a company as large as Veikkaus, with an expected of over 300 million unique user sessions per year, the sheer amount of data generated may be prohibitively—or at least pointlessly—expensive to track. As such, a lower sampling rate, such as 10%, may be sensible.

Much of an APM's real benefit comes from the fact that it gets its data from real customers in the production environment. While Veikkaus uses both extensive manual and automated testing, such tests can never produce full coverage and are often dangerous or impossible to perform outside specific test environments which may not always reflect the real state of production. This keys into *web performance analytics* and *alerting, error detection and error analysis* as described below.

2.4.3 Synthetic Monitoring

In addition to *passive* real-user monitoring, APMs use active synthetic monitoring as well: simple scripts to perform specific actions or calls to supplement the APM's RUM capabilities. At their simplest, synthetic monitoring could be something like periodically calling a specific API to ensure the service was still up; more complex test automation scripting to simulate series of user actions are also possible.

Synthetic monitoring is used to supplement real user monitoring, allowing one detect problems even during low-traffic periods or to, for example, test the performance of services that are as of yet unavailable to the general public.

2.4.4 Server/Application Monitoring and Performance Analytics

APMs are able to collect information and performance data on the hosts, containers and applications it is currently monitoring, ranging from CPU usage to round-trip response times. They monitor connection status in real time, whether by real user monitoring or synthetic monitoring as defined above. This data is used both for trends and analytics, but also for alerting should problems arise or should performance degrade too much.

2.4.5 Web Performance Analytics

For the most part, APMs offer fairly extensive web performance analytics tools, capable of measuring page load times by standard web vitals measurements such as First Input Delay and Largest Contentful Paint [Web20] [Zie20]. Users may be filtered based on their location, their browser, their OS and other such rules to determine, for example, how well the page performs in a specific browser type.

Discovering performance anomalies is quite crucial. The book *APM in the Digital Enterprise* argues that performance issues measured in fractions of seconds can both translate into significant losses and negatively impact customers' perception of brand quality—and that those issues are often not even discovered until the customer loss has already occurred [CPS17]. An APM would be a useful tool to discover such issues and combat them in a timely manner.

While these sorts of measurements are available through a large variety of free and commercial tools, an APM ties them into the rest of its full-stack monitoring capabilities. Should a page load be delayed due to a slow API call for example, one should be able to follow the trace and pinpoint the exact service or method that caused the delay. This, combined with APM's Real-User Monitoring capabilities, is a significant advantage compared to most other solutions.

2.4.6 Alerting, Error Detection and Error Analysis

Real-time alerting is a key aspect of all fully featured APM tools and one of their main selling points. Should a service be slow or unresponsive, or should API errors or JS

exceptions happen, an APM is able to alert its users of the fact, either through its dashboard or through other sorts of notifications.

These alerts are based on either manually defined baselines, such as pre-set maximum API roundtrip times, or dynamic, prediction based AIOps anomaly detection, such as a sudden, unexpected slowdown in a database server despite the number of requests made towards it being the same as before.

As APM features full-stack monitoring, they aim to be able to not only tell that something is amiss but also to pinpoint the source of the problem: they are not able to merely detect errors, but they also attempt to *analyse* them. Should a real user have issues with placing a bet on Veikkaus.fi, the APM should be able to identify the exact problem.

Suppose the connection to the database server is down and no requests are getting through; the APM would easily be able to pinpoint the problem by following the request and showing that the server is not responding to the query. Or, perhaps due to a bug, an ongoing sports event is marked as being finished in some backend service somewhere and users are prevented from placing bets on it: the APM would be able to display the exact service where the request stops, the exact error code it gives, even the exact stack trace of the service and the method from which the error was given from. While an APM is not guaranteed to get to the bottom of the real problem, it should be of some aid in at the very least locating it.

2.4.7 Automatic Topology Discovery and Mapping

A key feature of the major commercial APM tools is automatic topology discovery. By injecting their agents into the codebase, they are able to monitor relations between hosts and containers, between different application components, a real-time, automatically updating map between them is drawn. This not only helps in visualising the application itself, but also serves as a key monitoring hub for the APM itself to report on traffic and potential problems.

As of now, all four major commercial APM tools included in this report offer a view like this, whereas no open-source solution has a fully functional equivalent: while views like this are possible to manually construct in many other applications, automatic topology mapping is quite unique to APMs.

2.4.8 Business Transaction Management

All APM tools are also business transaction management tools, allowing one to define custom business transactions, coherent sets of user actions or use cases. For example, one could define ‘user accesses Veikkaus.fi -> user logs in -> user navigates the Pitkäveto page -> user places and confirms a bet’ as a single business transaction to be analysed and dissected further. One could, for example, measure how much page load times affect the likelihood of a user following through with the transaction, or with some custom code, even how much they affect the size of their bets; or should the Pitkäveto page fail to properly load, for example, due to a failure of a third-party service to respond, one can measure the error’s monetary impact.

With the ability to create custom dashboards for specific users, featuring the specific transactions they are interested in, this is a feature that may be of some use even to non-technical people who wish to monitor, for instance, the volume of sports events being bet on at any given point in time.

2.4.9 User Path Tracing

Modern APMs do not just sample data, they fully record individual user sessions—in some cases, down to the exact mouse movements they make. The data, should it be stored, can be accessed at a later time to get an exact view of what the user did and what kind of requests were made: for example, exactly 5.48 seconds into their session, they clicked a specific link, which generated a specific call that one can further drill down to analyse further.

Some tools even allowed replaying an end user’s every action so that they may be re-played back exactly by the APM [DyS20]. This may be helpful in redoing the exact steps that generated a specific error. Users can be filtered down based on browser type, country or even Veikkaus user ID or username. As such, in theory, should someone call Veikkaus Customer Support, one would be able to replay back their exact session just by knowing their username and find the exact problem they were having. Naturally, though, this is only actually viable if 100% of customer sessions are being traced. While every major APM tool offers the feature, the storage might get expensive; as such, lowering the sampling rate to an arbitrary lower value is also possible.

2.4.10 Extensibility

All major APM solutions are quite customisable. They feature the ability to both import custom data from external tools and applications (through custom plugins), and they all support an outwards-facing, OpenTracing API for exporting data as well. In theory, this means that even if the APM is unable to replace some tools currently in use, they should be able to display vital data received from them on their own dashboard, or even include it in their own analytics.

2.5 Tools Currently in Use

Commercial APM tools are large enterprise solutions that come with a multitude of interworking tools. Furthermore, all major APMs come with ways of integrating them with third-party tools, both in the sense of receiving data from them, as well as outputting data for them, such as by the means of an API. As such, when it comes to the needs of Veikkaus, it's important to not only consider the APM in a vacuum, but also to take into account the real environment it is to be deployed into.

Some value could be gained from an APM if it were able to replace tools currently in use. In the case of commercial tools specifically, this could be a cost saving measure, as tools currently being paid for could be retired. In addition, there is some value merely in simplifying the application ecosystem currently in use by Veikkaus; the current situation is a somewhat complex one with a wide array of sometimes functionally overlapping applications in use by various teams within the company and being able to get rid of at least some of them in favour of an APM would be at least of some help in simplifying the matter.

Some tools with overlapping functionality with an APM have been identified as follows:

Icinga: Icinga is an open-source network monitoring tool, able to monitor network services and hosts, collect performance data and generate alert in the case of outages [Ici20]. As of now, Icinga 2 largely forms the backbone of Veikkaus's network and host monitoring, with the ability to see the network at a glance and to get swift alerts should, for instance, a database server go down and stop responding to requests.

The overlap between APM and Icinga is obvious: there is nothing that Icinga does that an APM should not be able to do. The synthetic monitoring capabilities, performance data gathering and alerts are all features that each APM on the market are capable of. In theory,

an APM better able to detect problems due to its real user monitoring capabilities. Icinga is able to alert if a server is not responding to its calls quickly enough, but it might not see anything wrong in a misconfigured server that a critical service cannot connect to due to a firewall rule, or one that is erroneously sending malformed data that cannot be parsed in its destination. APM tools promise the ability to do just that, as they should be able to report on any errors that an end user of the web service sees, no matter what their cause—while perhaps assisting in finding that cause. In that, Icinga is likely to become redundant after an APM tool is implemented.

Grafana: An open-source analytics tool, Grafana collects data from a variety of sources, provides visualisations for them and allows its users to set up custom alerts [Gra20]. Grafana is currently used by Veikkaus to primarily aggregate and visualise performance data.

As with Icinga, APMs promise to do everything Grafana already does for Veikkaus: collecting performance data and helping to display it. In addition, the anomaly reporting functionality of APMs promises to be more advanced than Grafana's own: while you might be able to set alerts Grafana in the case of performance dipping under satisfactory levels, modern APMs tend to feature AIOps capabilities that automatically detect such anomalies and identify their root cause without a need to set up such things manually.

Telegraf and InfluxDB: Responsible for gathering the performance data that is then visualised in Grafana, Telegraf collects data from a variety of databases, cloud services, containers and servers [InA20]. InfluxDB is a time series database that Veikkaus uses to store the data gathered by Telegraf [InB20].

As APM tools come with their own agents to gather data and have their own ways to store it, neither Telegraf nor InfluxDB should serve any further purpose after an APM is implemented.

Prometheus: An open-source time-series database and data visualisation tool that largely fulfils the same function as InfluxDB does, having Grafana support for visualisation [Pro20] in use by some groups within Veikkaus in place of InfluxDB. It should be replaceable by an APM for similar reasons.

SpeedCurve: SpeedCurve is a front-end performance measurement tool that provides on such things as rendering speed and user engagement [Spe20]. It is largely used by front-end developers of Veikkaus to get feedbacks on page performance.

All commercial APMs feature similar capabilities to measure performance, with the added benefit of more granularity, with up to 100% of real user sessions recorded for further analysis, if so wished.

Hotjar: A user behaviour visualisation tool, Hotjar's primary use is in generating heat maps based on how users navigate on a website, as well as recording user sessions [Hot20].

The real-user monitoring capabilities of APMs have significant overlap with Hotjar, though the specific function to generate heatmaps may not be found in every APM tool. Depending on the capabilities of the APM chosen and the needs of Veikkaus regarding user behaviour visualisation, Hotjar may or may not be needed after an APM tool is implemented.

Cloud Platform Specific Tools: AWS and GCP come with a variety of monitoring tools, such as Amazon's X-Ray and Cloudwatch and Google's Stackdriver. These are unlikely to be replaced by an APM, but as all modern APM tools have support for integration for both cloud platforms, an APM might be used in the future to visualise some or all of the data gathered by the above tools, easing the process of monitoring the environment as a whole.

3 Requirements Gathering & Analysis

3.1 Initial Considerations

There are certain things that need to be taken into account when it comes to choosing a commercial off-the-shelf (COTS) product that would not necessarily come up when developing your own software. After all, though the APM field has grown significant in the recent years, there are still only a limited number of APM providers capable of providing a product that fits within Veikkaus's needs and specifications. For one, the requirements gathering process has to be somewhat curtailed; they have to be grounded in reality and need to be tailored to fit the products that already exist on the market.

Functionality and reliability stand as the most important aspects when it comes to choosing a COTS product [TiK05]. Out of those, functionality is the easy one to focus on. The most obvious way to compare products is to see what functionality they have that their

competitors do not. Reliability should be considered as well, but only as far as it can be phrased into discrete requirements. Meanwhile, a quality such as “ease of use” is too subjective to be reasonably evaluable through a mere requirements gathering process: comparing the products on the market from such a perspective would require an entire process of its own with all the tools available to use for comparison’s sake, which was not a possibility for this analysis.

Another important thing to consider was Veikkaus’s status as a government monopoly. As such, there are some legal considerations regarding purchases it makes. Products above a certain price need to be put through a process of *tendering*, and given the size of Veikkaus’s ecosystem, there is a very real possibility that it will be expensive enough to have to go through it. If that will be the case, the gathered requirements will have to double as an RFI (request for information) document—something the vendors themselves will respond to and verify, and something that can be used to score products competitively.

As such, requirements that no vendor could possibly satisfy with their product are not only useless, but potentially an active hindrance to the goal of acquiring an APM. Should the gathered list be used as an RFI, in the worst-case scenario, a mandatory requirement phrased in such a way that no vendor could possibly fulfil it would essentially put a pause to the whole tendering process as no offers could be made.

3.2 Requirements Gathering

Getting a list of requirements for the APM tool was one of the primary purposes of the whole project. Not only it would initially help Veikkaus to look the right way and to examine the right sorts of tools and to compare the APM products between each other, but it would also be of use in getting the actual offers from vendors. The requirements gathered by the elicitation were categorised into broad classes, and they were split between *mandatory* and *optional* ones. Any APM tool that would be considered would have to satisfy, at minimum, the mandatory requirements. The optional features would be where the actual comparison between them would happen; in later interviews, they would be scored from 0 to 100 based on perceived importance to weigh them between each other.

3.2.1 Initial Elicitation

The initial requirements were gathered from three primary sources. Firstly, as this was

not Veikkaus's first attempt at an APM acquisition, some old requirements still existed that were gathered during 2017 - 2018. They served as an initial basis for the new requirements, though with the changed—and changing—infrastructure some of them were too outdated to be used as-is. Secondly, the author's own understanding of what an APM is and what it is able to do was used to create some new requirements and adjust old ones.

Thirdly and finally, initial interviews were performed with five of the six interest groups as detailed in chapter 2 of this paper: front-end development, performance testing, infrastructure, networks and production support. The sixth group, back-end development was brought in at a later time to help answer questions from a development point of view that the front-end development team felt incapable of accurately responding to.

This stage of interviews was little more than an informal chat, lasting between 15 to 30 minutes, where the author sat down with a representative of each of the five groups to understand the work they did and to understand their point of view in the context of Veikkaus's infrastructure as a whole. With those unfamiliar with APMs, the author talked to them about what the tool did and how it might help their work. Finally, their opinions, wishes and fears regarding the tool were used as a basis for some preliminary requirements.

3.2.2 Refinement

Talking to a single representative per group alone risked providing a one-sided view, and as such, larger group interviews were conducted with the initial five groups. Between three to seven people were present in those five interviews, each lasting roughly an hour, with everyone having a chance to ask questions and pitch ideas.

This second stage of interviews proved to be quite fruitful, with multiple, mostly optional, requirements gathered based on questions and ideas posed by the interviewees. At this stage, the list of requirements was mostly finished, albeit unprioritized and unscored.

3.2.3 Individual Scoring

The requirements had to be split into mandatory and optional ones, with the optional ones given a weighed score based on their perceived importance. The third stage of interviews was about just that. With a single representative from each interest group—now including the sixth one, back-end development—each requirement was discussed one by one in six different interviews. If the interviewee was not confident in their ability to accurately

evaluate a requirement, they were permitted to give it a dash (—) as their score to not have it impact the final average. Otherwise, they would give it a score between 0 and 100 to represent their personal take on how important they considered the requirement to be.

A few requirements were adjusted or merged based on the suggestions of the interviewees, and finally, at the end of this process, the list had more or less taken its final form. However, for certain requirements, opinions between the groups varied massively, with some optional requirements being rated extremely high by some and extremely low by others. It was clear that merely taking an average would be insufficient. A final interview would be needed to form a consensus.

3.2.4 Group Scoring

The fourth and final interview stage consisted of just one interview, gathering together all six who had been involved with the individual scoring. In a single, long, round-table discussion, each requirement was once again individually gone through, with a calculated average score based on the results of the previous interview stage displayed for each requirement.

With some requirements, the average was deemed sufficient to settle on, but other ones elicited discussion and debate around the table and something very different ended up being finally settled on. Ultimately, some requirements that had been deemed extremely important (with a score of 90 or above) were moved from optional to mandatory as the consensus of their importance was clear. The whole interview lasted for just over two hours.

3.2.5 Final Considerations

Though not involved in the interview process per se, the requirements were finally looked through by Veikkaus's information security and data protection personnel to ensure nothing was missing. A couple of extra additions were made to ensure that GDPR-compliancy was properly addressed in the requirements listing. In addition, a cross functional virtual forum of IT professionals at Veikkaus known as the 'Technology guild' further vetted the list to confirm its validity and to ensure that there were no obvious shortcomings.

3.3 Categorisation

In the paper *User Requirements Analysis: A Review of Supporting Methods*, requirement

categorisation is roughly split into *user requirements*, *usability requirements* and *organisational requirements* [MaB02]. While the paper is not explicitly designed with COTS requirements analysis in mind, the categorisation was deemed universal enough to fit this analysis as well. Out of those, user and organisational requirements were the more relevant categories, useability requirements less so. For the most part, the usability of the tools is difficult to measure without being able to use them first-hand and given the limited field of actual available products to work with, many useability-related requirements would have been unrealistic. Still, some quality-of-life related wishes were made by the people interviewed that could be turned into relevant requirements.

To further organise the gathered requirements, they were split into eight subcategories, as outlined below:

User Requirements:

- Analysis: Requirements pertaining to analytical capabilities of the product, including both their ability to gather and display relevant information. Special consideration was given to RUM-related requirements, as some interviewed interest groups found them considerably more important than others.
- Monitoring: The product's capability to automatically notice anomalies and report on them. Of high value to support personnel within Veikkaus, less so for others.
- Technologies: Individual features that were deemed important by those interviewed, pertaining to specific supported technologies.

Usability Requirements:

- Maintenance: Requirements pertaining to ease-of-use in regard to installation and maintenance from the point of view of Veikkaus employees.
- Support: The ease and availability of support for the product.

Organisational Requirements:

- Legal: The legal requirements were mostly specific wishes by Veikkaus's information security and data protection team and dealt with European Union law compliance.
- Licences: Some fears were raised regarding the licensing terms of the software. Related requirements were given their own category.

- Performance: Given the size and expense of Veikkaus's online infrastructure, performance is critical to the organisation as a whole. Not only would a poorly performing APM be mostly self-defeating in purpose, it would also be a potentially massive expense.

3.4 Analysis

Though the individual scores given by the various interest groups tended to vary wildly, some obvious trends could be found (see Appendix 1). All groups placed high weight on the analytical requirements of APM, particularly regarding back-end monitoring capabilities, such as automatic error detection and display. The RUM aspects were somewhat more divisive, though that is perhaps no surprise. The importance of collecting browser metrics (req. O11) was rated very high by both front-end and back-end development (80 and 100 respectively), but much lower by those working at infrastructure and production support (50 and 40). An opposite trend could be found when it came to maintenance related requirements. Automatic installation into containers (req. O19) was rated a full 100 by the infrastructure team, but only 50 and 0 respectively by front-end and back-end development teams.

In general, a trend could be seen that the various support groups tended to value the requirements based on their own field of work, which was what made the fourth stage of the requirements analysis process so vital. If not for the final stage of discussion to find a consensus, many scores would have looked quite different from how they are now. Requirement O19 averaged at only 63 points based on individual scores, but after a round-table discussion, it was collectively deemed as important enough for its final score to be bumped up to 80. The development groups had not considered the sheer workload it would take to manually install an APM agent into every deployed agent in their own consideration but ended agreeing with the prevailing opinion when they had an opportunity to talk to others.

There were a handful of upsets. Though a feature heavily featured in vendor marketing materials, the 100% RUM sampling requirement (O8) was deemed unimportant by most—a nice extra feature, perhaps, but not something most imagined themselves finding much use of. The business dashboard feature (O10), meanwhile, was somewhat controversial: with scores ranging from 0 to 80 and two groups declining to answer, there was no easy consensus to be found. It was not considered to be a useful feature to any of the

persons involved in the requirements analysis process, though some were of the mind that it might be useful for Veikkaus employees involved in marketing or business side of things.

For the majority of the requirements, though, the final score was very close to the average of the individual scores gathered. Though there was a lot of discussion during over two-hour process, consensus was quite easily found for most of the scores as soon as everyone had had the chance to weigh in.

4 APM Comparison

For the purposes of this study, to keep the scope relatively limited, eight APM tools were initially chosen for comparison: four commercial APM tools and four open-source alternatives.

4.1 Products Considered

4.1.1 Commercial APM Tools

Out of the commercial ones, *Dynatrace* and Cisco's *AppDynamics* were chosen partially because Veikkaus had some experience with both from an earlier attempt at APM acquisition, but primarily because they were deemed current market leaders in Gartner's analysis [FeC20]. *New Relic APM* was likewise one of current market leaders, with a feature set and ability to execute that was very comparable to the other two.

Datadog APM was chosen for a comparison's sake for an opposite reason: classified as a *Visionary* rather than a *Leader* in Gartner's Magic Quadrant, Gartner's assessment of them signifies a low price and a feature set very comparable to the current market leaders, while cautioning that some of its features are less mature than its competitors'. As such, it served to both provide a potentially cheaper option while also acting as a baseline for comparison. The three other commercial options were clear market leaders, but from the perspective of Veikkaus, it was unclear whether there truly was a major difference between a market leader and a slightly less developed product. In addition, some of Veikkaus developers had previous experience with Datadog, albeit only from earlier days when it focused on cloud infrastructure monitoring and was not yet feature-rich enough to be classified as a full-fledged APM tool.

A potential risk when it comes COTS purchases is the possibility of obsolescence: the end of a product's life cycle after purchase, which puts a stop to updates and support. Though there can be any number of reasons for obsolescence, they often stem from decrease in product popularity, corporate takeovers or similar market factors; regardless of the cause though, they tend to be hard to predict [Mer06]. When it comes to the four products chosen, though, they all have well-established user bases and commercially profitable companies behind them, with seemingly solid product roadmaps for the future. Thus, choosing from amongst them is much less risky than trying to aim for a newcomer into the market.

The two major exclusions from this comparison were Broadcom's DX APM and Splunk's SignalFX APM. Based on Garnet's assessment, Broadcom's APM was somewhat behind the top three competitors of Dynatrace, AppDynamics and New Relic while not providing a notably cheaper pricing model. SignalFX meanwhile was quite similarly positioned as Datadog, in a clear underdog position compared to the current market leaders, without having anything else to obviously set it apart.

Likewise excluded were all APM tools on the left side of Gartner's Magic Quadrant, rated low on completeness of vision. With the scale and importance of Veikkaus's gaming systems, it was deemed unlikely that they would satisfy all the requirements Veikkaus would end up having.

4.1.2 Open-Source Alternatives

One of the main purposes of the study, initially, was to determine whether an open-source solution would be a viable alternative to the major commercial tools. If that were the case, there would be room for massive savings, as on the scale of Veikkaus, commercial tools could easily cost hundreds of thousands of euros, or even more, annually. The four tools chosen for comparison were *Pinpoint*, *Sentry*, *Elastic APM and* Apache's *Skywalking*, each chosen based on both their large, frequently updated GitHub repositories as well as community recommendations.

Unfortunately, it soon turned out that the open-source tools fit neither the minimum requirements of Veikkaus as set by the interview process nor the three-dimensional model for an APM as set out for Gartner. The four open-source tools inspected seemed strictly aimed towards distributed tracing and cloud monitoring above all else, lacking aspects such as business analytics and front-end monitoring. As such, under the definition of APM

used for this study, they did not qualify as APM tools to begin with; more crucially, they did not satisfy the minimum requirements, in particularly regards to RUM analytics.

As such, the author had to conclude early on that no open-source tool that could be identified would suit the needs of Veikkaus, and no further consideration was given to them.

4.2 Comparison Process

With the open-source alternatives filtered out early on, further examination was limited to the four commercial APM tools selected for analysis. Going through every gathered requirement one by one, each product was examined to see if they were a fit. Actually getting the answers was a non-trivial task, however. While the documentation and product pages for the APM tools provided many answers, not everything was answerable by them alone. Sometimes, the requirement was too specific or niche to be answered directly by the product's documentation; occasionally the answer found was unclear or inadequate. In general, trusting on a single type of a source might lead to inaccuracies; it was clear that multiple different information sources would be needed.

When it comes to picking information sources for gathering data on COTS products, Ayala, Botella and Franch, in their paper *On Goal-Oriented COTS Taxonomies Construction*, discuss several types of information sources they themselves used as a basis for information-gathering for their own COTS analysis process [ABF05]. Their categorisation was a useful basis to begin considering potential sources of information for this task as well, but not all sources they used were useful for this work. Namely, very little in terms of official standards or taxonomies currently exist for the APM domain: the only apparent authoritative source that could be found was Gartner. Secondly, there was no opportunity to test the various APM tools in practice. Four distinct categories of information sources were deemed suitable as a basis for this work:

- **Vendors Information**: The vendors themselves marketed certain aspects of their products heavily on their websites, often with images or videos to demonstrate features. These proved to be a valuable source both for the initial research and for subsequent comparisons between the examined products.
- **Domain Descriptors**: Few actual papers have been written on APMs, and virtually none that talk about the specifics of their features. Gartner's yearly analysis on the APM market, however, did provide useable insights. In addition, a number

of third-party websites featured discussion about the products that was used to verify certain claims.

- **Test of Tools and Systems**: While the author did not have the opportunity to try all four tools first-hand for the purposes of this paper, but each product had an online user's manual that provided useful information about more their specific features that the marketing materials themselves were too generalised to provide.
- **Oral Information**: Ultimately, the only way to gain answers to more specific questions was to ask from the vendors themselves. To get answers to many specific questions, the author contacted the companies directly as a representative of Veikkaus. Following a series of email exchanges, as well as video conferences through Zoom, Microsoft Teams and BlueJeans, most of the remainder of the questions regarding the requirements were answered.

Some requirements, however, were deemed difficult or impossible to answer before the product could actually be seen in use. This was the case for two requirements: "The product must not hinder or prevent any components of Veikkaus gaming systems from working." and "The product and its agents must be installable and made ready within one working week."

As much as the vendor might promise in an email that that their product will not cause any hindrances in Veikkaus's production environment, such claims are impossible to fully verify without seeing them in use, and any possible future proof of concept demos that may ensue fall outside the scope for this thesis.

4.3 Results

Out of a maximum of 1910 points, the final score for Dynatrace was 1650, while New Relic and Appdynamics, with their essentially identical feature lists, both had a score of 1560. The results of the comparison process support Gartner's ranking of the respective tools. Dynatrace, AppDynamics and New Relic are each very comparable to each other, with Dynatrace having an edge by a very small margin.

Datadog's performance would have been adequate if not for it failing to satisfy certain minimum requirements. With its non-EEA based support team and its lack of fine-tuned user role control, it was deemed insufficient for the purposes of Veikkaus as it currently exists.

There is a risk for the above table to be slightly misleading. While all three main tools largely satisfied the same requirements, it does not mean the tools themselves are identical. There may be major differences between the tools regarding user experience or additional features that fall outside of Veikkaus's requirements. In addition, the comparison process was very binary, concerned only with whether the tool adequately satisfied the requirement or not. For instance, if one product has significantly better automatic error analysis than another, it is not shown in the above table: the only thing that it shows is whether the product has the feature or not.

In addition, despite the similar ranking of the three main contenders feature-wise, there is a drastic difference in the logic of how the products are priced. Where AppDynamics for instance has cheaper prices for test environments compared to production ones, Dynatrace does not. And where both of them price themselves based on the number of hosts monitored, New Relic's pricing is based on the number of users using the APM tool, not the number of hosts involved. As the scope of this work is limited to the needs of Veikkaus regarding the capabilities of APM, the exact details of pricing fall outside of it.

5 Discussion

5.1 The Research Questions

This work's primary purpose is to serve as a look into Veikkaus's current situation in regards to its network infrastructure, as well as a guideline for choosing the right tool (if any) to supplement any holes or fix any flaws within it, and the research questions were all chosen to revolve around it. Even in retrospect, they were logical choices that stemmed from very real needs and served to both focus and structure the work.

5.1.1 RQ1: What are the various requirements regarding APM that different interest groups with Veikkaus have?

As initially anticipated, the first research question was the most complex one to answer. Due to the size of Veikkaus's IT personnel, there were many different deficiencies, many different problems and many different expectations for better tools. It helped, though, that the work focused solely on problems than APM could solve: though the state of the current APM tools on the market is impressive, they can't do everything, and that made it possible to filter out requirements that no APM could possibly answer, narrowing down

the work involved with answering the research question.

Ultimately, only a small handful of people interviewed were already familiar with APMs, requiring the author to explain their functionality and features. This, no doubt, affected the requirements gathered from those interviewed: they did not spring entirely organically, but were specifically given in the context of the person's expectations around what an APM might be able to do. As a positive, it helped to keep the interviews more focused; as a negative, it may have discouraged out-of-box thinking that could have uncovered problems that might have been solvable with non-APM tools and other methods.

The gathered list ended up being entirely satisfactory, however, with no errors or significant shortcomings caught during later stages of the work.

5.1.2 RQ2: Are any of the above requirements solved easier by some other means?

The second research question was important for two reasons. Firstly, it was important to consider whether an APM was necessary *at all*: as expensive enterprise tools, it would have been better if their functionality could be satisfyingly replicated with a variety of cheaper, perhaps even open-sourced solutions. Secondly, as an additional benefit, if any requirements had turned out that an APM was simply not an answer to, it would have provided some extra value to identify and answer those requirements on the side—perhaps not as a part of this work, but as something to be addressed through different means.

Ultimately, APM did prove to be a good match to most of the requirements gathered, and a brief analysis of non-APM tools on the market revealed shortcomings compared to an APM solution—specifically, the real-user monitoring aspects and the analysis that would integrate genuine real end user data from every level of the stack was not something one would get with non-APM solutions.

5.1.3 RQ3: Which APM tool is the best fit for Veikkaus?

The third research question was always the original purpose of this work, albeit one that would have been pointless to answer if the other two did not provide the basis for it. For years now, there has been will within the company to acquire an APM for network monitoring and testing purposes, but due to the complex and evolving market and the high price point of the tools involved, no-one wanted to act rashly when it came to making a decision.

As of 2020, Gartner Research identifies fifteen different APMs on the market [FeC20]. If one wishes to broaden their definition of an APM from Gartner's own, rather strict one, there could be an arbitrary amount more. There's no single obvious way to choose between them. One could. perhaps, simply trust Gartner's own Magic Quadrant analysis and pick the one they deem the best; or, one could look at the user bases of the APMs and choose the most popular one, expecting most support for it. Neither of those approaches would have specifically matched the tool for Veikkaus's needs, though, and there would have been a very real risk of getting something overly expensive, or something focused on different things than what Veikkaus really needs.

As such, it made sense to go through every tool on feature-by-feature basis and see what the differences were between them—and how those differences measured up to the gathered requirements. As it wasn't feasible to do that for every APM on the market, Gartner's analysis was used as a template for identifying the market leaders, as well as a single non-market leader to give more of a baseline.

The relative homogeneity between the tools came as a slight surprise, if largely a positive one. Multiple APMs satisfying the same requirements gives Veikkaus more free hands in choosing between them and perhaps negotiating a price.

5.2 Review of Methodology

As this work was one primarily revolving around gathering requirements, the first question was one of how the process would be done. Several techniques, such as user story gathering and UML modelling were considered, but the unorthodox nature of the problem this study was meant to answer made it difficult to apply specific methodology. The three key issues were as follows:

1. As the purpose of this study was to help choosing a commercial off-the-shelf tool from a narrow field of products, the gathering process would have to be somewhat closed and curtailed with requirements that no product could possibly satisfy having to be dismissed off-hand.
2. The people interviewed did not themselves know much about APMs and how to use them, so they could often not express clearly what they wanted the product *to do*—merely what problems they had that needed a solution. Though the author could have helped them formulate their requirements as user stories or use cases

for the context of APMs, it would have been dangerously leading.

3. Veikkaus wished for the final list to be useable as a basis for potential competitive tendering, which meant that requirements had to be specific and answerable with a yes or no question of whether the product satisfied a requirement or not. At the same time, they couldn't be too specific, or they might accidentally unfairly favour one product over another.

A review of literature did not provide a single comprehensive method for gathering, analysing and weighing requirements under such constraints. As such, a more informal process was used, one that had been already applied in-house beforehand for past competitive tendering projects: multi-stage interviewing and round-table discussion to reach a consensus.

With that in mind, the requirements themselves were somewhat "soft", reached through discussion and gentle prodding. A viable alternative would have been to first comprehensibly introduce the interviewees to the specifics of what an APM is meant to do and how they work, and then gather specific user stories in response from more well-informed sources. This would have made answering RQ2 more difficult, however, as the interviewees would have likely given every answer specifically through the lens of an APM and nothing else. It might have created blind spots in their answers and stopped them from bringing up issues that weren't strictly APM related; on the other hand, they might have been able to give more granular, specific requirements that would have helped set the APMs further apart from each other. Perhaps with such a method, with more fully informed interview subjects, we would have seen more distinct differences between Dynatrace, AppDynamics and New Relic as products.

5.3 Limitations of the Study

The study was conducted with a strong focus on Veikkaus's own needs, albeit with an expectation that it would provide some insight into the current state of the APM market. In some ways, this was a success: in particular, the fact that the market leaders were so close together feature-wise was an insight that was by no means obvious until a comparison was performed with the help of the list of requirements. At the same time, though, it's important to keep in mind that only four APMs were seriously compared in this study. The actual field is quite a bit larger, and a full feature-by-feature comparison of every

APM tool in the market would be interesting to see. Gartner Research's yearly analysis is the closest equivalent, but it lacks the granularity of diving deep and specifying what one tool can do that another cannot.

Similarly, RQ1 could have been expanded upon. APM is not interesting only to IT person el, but leadership and marketing as well due to its business transaction management side and the potential monetary value it provides by analysing customer behaviour and catching errors earlier. They might have been able to provide different, less technical sorts of requirements and thus shifted the focus of the study towards looking into even broader applications of an APM.

As mentioned in the section above, a more robust sort of methodology could have been discovered and applied to RQ1, though no obvious candidate ever seemed to spring up during the course of the study. Similarly, while RQ2 was given its own consideration, some clearer data could have been gathered on non-APM tools that would have fulfilled similar functionality. In that way, a proper comparison could have been conducted to show exactly which features of an APM would have been replicable with non-APM tools, and which ones were ones that were specific to APMs. Given the sheer scope of the field of network monitoring, traffic analysis, business transaction management, web performance analytics etc. though, it would have been a massive undertaking outside the scope of this work.

Finally, APM pricing was purposefully left out of consideration entirely. Not only would estimates have been difficult to make with vastly different pricing logics between the tools, but there's also some lack of transparency there from the APM vendors themselves. With a client as large as Veikkaus, it's likely that some tendering and negotiating would occur, making any possible 'listed' prices provided mere guidelines at best. Still, it would have been interesting to see if there really were large differences in prices, considering the relative homogeneity of the tools' lists of features.

6 Conclusions

In their book APM in the Digital Enterprise: Managing Applications for Cloud, Mobile, IoT and eBusiness, the authors argue that the growing complexity, containerisation and transition into cloud are making APM and UEM solutions essential tools for development [CPS17]. The general feeling amongst Veikkaus personnel interviewed for this thesis

seems to support the claim. The difficulty of understanding network topology in a microservices based architecture, the lack of capabilities to detect anomalies in production as they occur and the slowness in identifying their root cause were all singled out as major causes of worries, and all of them are seemingly addressable with a modern APM tool.

Though much of what an APM does is currently being done with a wide variety of tools, such as Icinga, Grafana and SpeedCurve among other, there are four key points that make APM a plain improvement over the current status quo:

The real-user monitoring capabilities of APM are not currently covered by any tool in use. If one user in twenty has troubles logging in at Veikkaus.fi due to an error in code or a misconfigured network device, the problem as it is now is unlikely to be caught until customers begin calling Veikkaus support. An APM should be able to pick up on those errors instantly and automatically.

As a single, full-stack observability solution, APMs are able to help draw conclusions that might be difficult with a set of miscellaneous tools. Where Telegraf and Grafana might be able to alert about degrading performance of a host, an APM might be able to find the exact JavaScript function that is causing the performance issues or might be able to link that issue with other errors happening in the service.

Some optional requirements, such as automatic topology mapping and AIOps based automatic anomaly detection are not covered by any other tool at the moment.

Being able to retire multiple data gathering, data storing and data visualising tools in favour of a single APM tool would simplify the current, complex application ecosystem.

As far as comparing the tools on the market goes, APMs, as enterprise tools that are meant to satisfy a wildly disparate set of monitoring and analysis needs, are difficult to objectively evaluate. With a company as big as Veikkaus, it is not easy to find consensus regarding what its needs are. The interest groups interviewed each had their own viewpoints and concerns, with quite different ideas regarding what the final product would and should do. The front-end development group, for instance, was particularly interested in its user experience monitoring and performance monitoring features, whereas those working with infrastructure and networks valued the visualisation and error detection features most highly. Thus, comparing the products to each other was not easy, though the open discussion and debate format of the last interview stage was ultimately very helpful, allowing the participants to compare and evaluate their own needs in comparison to others'.

Ultimately, though, there were no big surprises in the results. Gartner's initial analysis of the tools was more or less on the spot, with Veikkaus having no particular special needs that tilted the balance between the products. Dynatrace, the leading product in Gartner's own analysis, scored the highest, with AppDynamics and New Relic having only a marginally lower score. With such a small difference between them, the deciding factor in picking between them should likely be the price—assuming that nothing strange comes up in a proof-of-concept stage. As such, it is the author's recommendation that Veikkaus is to invest in an APM and choose one of the three above products, or potentially any other that satisfied the same requirements.

It should be noted that APMs are still a very developing field. There's constant ongoing innovation regarding new features. Even during the process of writing this thesis, the feature list had to be revamped and corrected, with Datadog for instance beginning to support single page applications halfway through this project, and New Relic changing their entire pricing logic in the summer of 2020 [Khu20]. It is likely that this work would be quite different had it been done a year earlier, or a year later. The comparison as it exists now is a snapshot of how things are at the moment of writing, and it may not hold true for very long into the future.

7 Sources

ABF05 Ayala C.P., Botella P., Franch X. (2005) On Goal-Oriented COTS Taxonomies Construction. In: COTS-Based Software Systems. ICCBSS 2005. Lecture Notes in Computer Science, vol 3412

Adr19 Jack A., What Is Application Performance Monitoring? May 23, 2019 https://www.appdynamics.com/blog/product/application-performance-monitoring/ [11.11.2020]

App20 AppDynamics, Auto Discovery and Mapping https://www.appdynamics.com/product/how-it-works/auto-discovery-and-mapping [11.11.2020]

Bal19 Ball M., New Relic Browser: Tracking User Experience with First Paint and First Contentful Paint Metrics July 18, 2019 https://blog.newrelic.com/product-news/browser-first-paint-first-contentful-paint/ [11.11.2020]

Blo18 Bloom J., How to Ensure Your Applications Meet Business Goals August

15, 2018 https://www.appdynamics.com/blog/product/ensure-applications-meet-business-goals/ [11.11.2020]

Cla13 Clay K., Forbes: Amazon.com Goes Down, Loses $66,240 Per Minute August 19, 2013 https://www.forbes.com/sites/kellyclay/2013/08/19/amazon-com-goes-down-loses-66240-per-minute/ [27.11.2020]

CPS17 Craig J., Pollard C., Sturm R. Application Performance Management (APM) in the Digital Enterprise: Managing Applications for Cloud, Mobile, IoT and eBusiness (2017). pages 89-90, 116-118, 174-175

Dyn20 Dynatrace, Business transaction monitoring https://www.dynatrace.com/platform/business-transaction-monitoring/ [11.11.2020]

DyS20 Dynatrace Support, User sessions https://www.dynatrace.com/support/help/how-to-use-dynatrace/real-user-monitoring/basic-concepts/user-session/ [11.11.2020]

FeC20 De Silva F., Rich C., Magic Quadrant for Application Performance Monitoring April 22, 2020 https://www.gartner.com/document/3983892 [11.11.2020]

Gra20 Grafana Labs, Grafana https://grafana.com/grafana/ [11.11.2020]

GSR19 Ganguli S, De Silva F., Rich C, Market Guide for Digital Experience Monitoring September 5, 2019 https://www.gartner.com/document/code/384300 [11.11.2020]

HHM17 Heger C., van Hoorn A., Mann M., Okanović D., Application Performance Management: State of the Art and Challenges for the Future. In *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering (ICPE '17)*. April 2017, pages 429–432.

Hir19 Hirschauer J., What Does APM Stand For? A Newbie Guide May 20, 2019 https://www.appdynamics.com/blog/product/newbie-guide-apm/ [11.11.2020]

Hot20 Hotjar https://www.hotjar.com/tour/ [11.11.2020]

Ici20 Icinga, About Icinga https://icinga.com/docs/icinga-2/latest/ [11.11.2020]

InA20 Influxdata, InfluxDB Platform https://www.influxdata.com/products/influxdb-overview/ [11.11.2020]

InB20 Influxdata, Telegraf https://www.influxdata.com/time-series-platform/telegraf/ [11.11.2020]

Khu20 Khurana M., Troubleshoot Your Stack, Not Your Monitoring Bill July 30, 2020 https://blog.newrelic.com/product-news/simple-pricing/ [11.11.2020]

MaB02 Maguire M., Bevan N. (2002) User Requirements Analysis: A Review of Supporting Methods. In: *IFIP World Computer Congress, IFIPAICT, vol 99.*, page 140.

Mer06 Merola L., The COTS software obsolescence threat (2006). In: *Fifth International Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'05)*.

TiK05 Tiwana A. and Keil M., Beyond cost: the drivers of COTS application value (May-June 2005). In *IEEE Software, vol. 22, no. 3*, pp. page 64-69

Pro20 Prometheus Docs, Grafana Support for Prometheus https://prometheus.io/docs/visualization/grafana/ [11.11.2020]

Spe20 Speedcurve https://speedcurve.com/ [11.11.2020]

Web20 web.dev, Web Vitals July 21, 2020 https://web.dev/vitals/#core-web-vitals [11.11.2020]

Zie20 Ziegelbecker T., More visibility into user experience with new web performance metrics and enhanced Visually Complete May 30, 2020 https://www.dynatrace.com/news/blog/more-visibility-into-user-experience-with-new-web-performance-metrics-and-enhanced-visually-complete/ [11.11.2020]

Appendix 1: Requirement Scoring

The gathered requirements were split into two categories: mandatory and optional. The optional requirements were weighed in importance and thus scored by the six interest groups interviewed: front-end development (Front-end), back-end development (Back-end), infrastructure (Infra.), production support (Support), performance testing (Perf.) and networks (Networks). The final score is not an average, but rather one reached by a consensus in a round-table discussion between the participants towards the end of the interview process.

Requirement	Class	Front-end	Back-end	Infra.	Support	Perf.	Networks	Final
Mandatory Requirements								
M1 The product measures the target environment's components' (such as Docker containers and hosts) performance (such as CPU load, memory usage and response times) and connection status in real time.	Analysis							
M2 The product is able to follow the target environment's systems' performance trends.	Analysis							
M3 The product is able to display the root cause of errors on code/SQL level on Java and database hosts.	Analysis							
M4 The product is able to discover the source of problems based on hostname or IP.	Analysis							
M5 The product is able to show 30 days of history based number of calls, response times and errors of different pages and services.	Analysis							
M6 The product is able to show the number of calls, response times and errors towards third party services originating from Veikkaus gaming systems.	Analysis							
M7 The product is able to display an automatic, real-time map of the data flow of Veikkaus gaming systems.	Analysis							
M8 The product is able to create a dashboard displaying a holistic view of the current status of Veikkaus gaming systems.	Analysis							
M9 The product is able to catch JavaScript exceptions and helps to identify their source.	Analysis							
M10 The product is able to catch HTTP errors and help identify their source.	Analysis							
M11 The product is able to separate the end customers based on their browser, device, location and connection type in its RUM analysis.	Analysis - RUM							
M12 The product must be able to black list specific information so that it doesn't get permanently saved in its RUM analysis.	Analysis - RUM							
M13 The product is able to make alerts regarding major changes and problems regarding response times and errors in different services within Veikkaus gaming systems.	Monitoring							
M14 It's possible to set a manual baseline for alerts and error reporting.	Monitoring							
M15 The product must not hinder or prevent any components of Veikkaus gaming systems from working.	Performance							
M16 The product must have the ability to set different role-based rights for its various users.	Usability							
M17 The product must support user or team specific views, such as custom performance dashboards.	Usability							
M18 The product must have the ability to prevent specific user roles from seeing specific target systems' information.	Usability							
M19 The product must support user access rights integration into Microsoft's Active Directory service.	Usability							
M20 The product and its agents must work properly in Veikkaus gaming systems without there being a need for manual configuration or manual changes to the program code.	Maintenance							
M21 The product and its agents must be installable and made ready within one working week.	Maintenance							
M22 The product must be useable in multiple different environments (such as production, staging and performance testing)	Licences							
M23 The product must come with an unlimited amount of user licences.	Licences							
M24 The product must have a web-based user interface.	Technologies							
M25 The product must work with all of the hosts and services used by Veikkaus gaming systems: Amazon Web Services, Google Cloud Platform, physical hosts and various virtual hosts.	Technologies							
M26 The product must support Docker and Kubernetes containers.	Technologies							
M27 The product must be loadable from a secure HTTPS endpoint.	Technologies							
M28 The product must have a clear product roadmap and the vendor must have sufficient resources for executing it.	Support							
M29 The product's support personnel who would handle any data must not be from outside of European Economic Area	Legal							
M30 No stored data may leave the European Economic Area.	Legal							
M31 Any stored personal information must be removable within 30 days.	Legal							
Requirement	**Class**	**Front-end**	**Back-end**	**Infra.**	**Support**	**Perf.**	**Networks**	**Final**
Optional Requirements								
O1 The product must automatically analyse errors and indicate them to the user of the APM.	Analysis	75	100	80	100	50	90	**85**
O2 The analysis history of the product must be browsable and searchable by the APM's user.	Analysis	50	50	50	30	80	40	**50**
O3 The product is able to display networking devices, such as routers and firewalls, in the gaming systems' data flow and is able to follow calls as they pass through them.	Analysis	75	100	90	70	70	100	**85**
O4 The product is able to track individual networking devices and list all calls that pass through it.	Analysis	75	100	90	70	20	100	**75**
O5 The product is able to identify high memory usage and slow executing code in its code level analysis.	Analysis	75	50	60	80	90	-	**70**
O6 The product has a customer path view which shows an individual user's path, the calls they have performed and the exact times for the actions they have taken while browsing Veikkaus.fi	Analysis - RUM	80	50	70	80	100	100	**80**
O7 The product is able to collect RUM-data with at least a 15% sampling rate.	Analysis - RUM	100	50	70	60	-	-	**70**
O8 The product is able to collect RUM-data with a full 100% sampling rate.	Analysis - RUM	20	10	80	30	-	10	**20**
O9 The user of the product is able to specify custom measurements and conversions and track them for its RUM analysis.	Analysis - RUM	50	100	50	50	-	-	**60**
O10 The user of the product is able to perform business-oriented dashboards from which one is able to see the monetary impact of performance changes.	Analysis - RUM	75	0	80	50	-	-	**50**
O11 The product has tools for site performance analysis, collecting basic metrics such as 1st paint, 1st user interaction and the loading times as reported by the end user's browser.	Analysis - RUM	80	100	50	40	80	-	**70**
O12 The product is able to use an automatic, dynamically updating baseline for alerts and error reporting (AIOps)	Monitoring	75	80	80	100	60	50	**75**
O13 The product is able to, through synthetic monitoring, occasionally make calls to specific APIs such as the Health APIs used by various Veikkaus services.	Monitoring	-	75	40	50	30	70	**50**
O14 The product has support for notifications for Slack and email.	Monitoring	15	100	80	100	70	100	**75**
O15 When the gaming systems are under a heavy load, the product must be able to drive its own services down and free resources for the gaming systems' use, such as by having a specific overhead cap.	Performance	20	50	30	40	80	40	**40**
O16 The product's configuration must be manageable through versioning.	Maintenance	15	50	100	-	60	90	**65**
O17 The product and its agents must be upgradeable centrally through a user interface.	Maintenance	25	100	0	80	60	70	**55**
O18 The product must have a single installable agent for all host types.	Maintenance	15	50	30	50	30	70	**50**
O19 The product must be automatically installable into new containers such as by a command in a Dockerfile	Maintenance	50	0	100	70	70	90	**80**
O20 The APM must be easily toggleable on or off.	Maintenance	20	0	100	90	95	80	**65**
O21 The product's pricing scales depending on real usage.	Licences	15	100	80	80	-	70	**70**
O22 The product's agents must be transferrable between various environments depending on real needs without any additional fees involved.	Licences	50	0	100	80	-	70	**60**
O23 The product must support integration into third party tools, such as by exposing an API for them to use.	Technologies	15	100	80	80	10	100	**65**
O24 The product must support importing custom data from third party tools.	Technologies	15	100	80	60	75	80	**70**
O25 The product must have a Data API that follows a common standard (such as OpenTracing, OpenCensus or Prometheus)	Technologies	15	100	60	100	60	80	**70**
O26 The product must have an installable native application.	Technologies	30	50	20	40	90	40	**40**
O27 The product's agents must be loadable asynchronously.	Technologies	35	100	60	-	-	-	**60**
O28 The product must be able to follow Single Page Applications, their internal events and their performance.	Technologies	75	50	50	30	-	60	**65**
O29 The product must support JavaScript source maps.	Technologies	40	100	60	90	-	-	**75**
O30 The product must have live support that is available immediately during the office hours of Veikkaus.	Support	25	100	70	70	50	70	**65**

Appendix 2: Vendor Comparison

An analysis of which products satisfy which gathered requirements. The performed comparison is strictly binary: the product either does or does not fulfil the requirement. In some cases, though, a footnote was warranted for further clarification. They will be listed below.

Requirement	Class	Dynatrace	AppDynam	New Relic	Datadog
Mandatory Requirements					
The product measures the target environment's components' (such as Docker containers and hosts) performance (such as CPU load, memory usage and response times) and connection status in real time.	Analysis	Y	Y	Y	Y
The product is able to follow the target environment's systems' performance trends.	Analysis	Y	Y	Y	Y
The product is able to display the root cause of errors on code/SQL level on Java and database hosts.	Analysis	Y	Y	Y	Y
The product is able to discover the source of problems based on hostname or IP.	Analysis	Y	Y	Y	Y
The product is able to show 30 days of history based number of calls, response times and errors of different pages and services.	Analysis	Y	Y	Y	Y
The product is able to show the number of calls, response times and errors towards third party services originating from Veikkaus gaming systems.	Analysis	Y	Y	Y	Y
The product is able to display an automatic, real-time map of the data flow of Veikkaus gaming systems.	Analysis	Y	Y	Y	Y
The product is able to create a dashboard displaying a holistic view of the current status of Veikkaus gaming systems.	Analysis	Y	Y	Y	Y
The product is able to catch JavaScript exceptions and helps to identify their source.	Analysis	Y	Y	Y	Y
The product is able to catch HTTP errors and help identify their source.	Analysis	Y	Y	Y	Y
The product is able to separate the end customers based on their browser, device, location and connection type in its RUM analysis.	Analysis - RUM	Y	Y	Y	Y
The product must be able to black list specific information so that it doesn't get permanently saved in its RUM analysis.	Analysis - RUM	Y	Y	Y	Y
The product is able to make alerts regarding major changes and problems regarding response times and errors in different services within Veikkaus gaming systems.	Monitoring	Y	Y	Y	Y
It's possible to set a manual baseline for alerts and error reporting.	Monitoring	Y	Y	Y	Y
The product must not hinder or prevent any components of Veikkaus gaming systems from working.	Performance	-	-	-	-
The product must have the ability to set different role-based rights for its various users.	Usability	Y	Y	Y	Y
The product must support user or team specific views, such as custom performance dashboards.	Usability	Y	Y	Y	Y
The product must have the ability to prevent specific user roles from seeing specific target systems' information.	Usability	Y	Y	Y	N[1]
The product must support user access rights integration into Microsoft's Active Directory service.	Usability	Y	Y	Y	Y
The product and its agents must work properly in Veikkaus gaming systems without there being a need for manual configuration or manual changes to the program code.	Maintenance	Y	Y	Y	Y
The product and its agents must be installable and made ready within one working week.	Maintenance	-	-	-	-
The product must be useable in multiple different environments (such as production, staging and performance testing)	Licences	Y	Y	Y	Y
The product must come with an unlimited amount of user licences.	Licences	Y	Y	Y	Y
The product must have a web-based user interface.	Technologies	Y	Y	Y	Y
The product must work with all of the hosts and services used by Veikkaus gaming systems: Amazon Web Services, Google Cloud Platform, physical hosts and various virtual hosts.	Technologies	Y	Y	Y	Y
The product must support Docker and Kubernetes containers.	Technologies	Y	Y	Y	Y
The product must be loadable from a secure HTTPS endpoint.	Technologies	Y	Y	Y	Y
The product must have a clear product roadmap and the vendor must have sufficient resources for executing it.	Support	-	-	-	-
The product's support personnel who would handle any data must not be from outside of European Economic Area	Legal	Y	Y	Y	N
No stored data may leave the European Economic Area.	Legal	Y	Y	Y	Y
Any stored personal information must be removable within 30 days.	Legal	Y [2]	Y	Y	Y
Requirement	**Class**	**Dynatrace**	**AppDynam**	**New Relic**	**Datadog**
Optional Requirements					
The product must automatically analyse errors and indicate them to the user of the APM.	Analysis	Y	Y	Y	Y
The analysis history of the product must be browsable and searchable by the APM's user.	Analysis	Y	Y	Y	Y
The product is able to display networking devices, such as routers and firewalls, in the gaming systems' data flow and is able to follow calls as they pass through them.	Analysis	N [3]	N	N	N
The product is able to track individual networking devices and list all calls that pass through it.	Analysis	N [3]	N	N	N
The product is able to identify high memory usage and slow executing code in its code level analysis.	Analysis	Y	Y	Y	Y
The product has a customer path view which shows an individual user's path, the calls they have performed and the exact times for the actions they have taken while browsing Veikkaus.fi	Analysis - RUM	Y	Y	Y	N
The product is able to collect RUM-data with at least a 15% sampling rate.	Analysis - RUM	Y	Y	Y	Y
The product is able to collect RUM-data with a full 100% sampling rate.	Analysis - RUM	Y	Y	Y	Y
The user of the product is able to specify custom measurements and conversions and track them for its RUM analysis.	Analysis - RUM	Y	Y	Y	Y
The user of the product is able to perform business-oriented dashboards from which one is able to see the monetary impact of performance changes.	Analysis - RUM	Y	Y	Y	N
The product has tools for site performance analysis, collecting basic metrics such as 1st paint, 1st user interaction and the loading times as reported by the end user's browser.	Analysis - RUM	Y	Y	Y	Y
The product is able to use an automatic, dynamically updating baseline for alerts and error reporting (AIOps)	Monitoring	Y	Y	Y	Y
The product is able to, through synthetic monitoring, occasionally make calls to specific APIs such as the Health APIs used by various Veikkaus services.	Monitoring	Y	Y	Y	Y
The product has support for notifications for Slack and email.	Monitoring	Y	Y	Y	Y
When the gaming systems are under a heavy load, the product must be able to drive its own services down and free resources for the gaming systems' use, such as by having a specific overhead cap.	Performance	Y	N	N	N
The product's configuration must be manageable through versioning.	Maintenance	Y	Y	Y	Y
The product and its agents must be upgradeable centrally through a user interface.	Maintenance	Y	Y	Y	N [4]
The product must have a single installable agent for all host types.	Maintenance	Y [5]	N	N	N
The product must be automatically installable into new containers such as by a command in a Dockerfile	Maintenance	Y	Y	Y	Y
The APM must be easily toggleable on or off.	Maintenance	Y	Y	Y	N [6]
The product's pricing scales depending on real usage.	Licences	Y	Y	Y	Y
The product's agents must be transferrable between various environments depending on real needs without any additional fees involved.	Licences	Y	Y	Y	Y
The product must support integration into third party tools, such as by exposing an API for them to use.	Technologies	Y	Y	Y	Y
The product must support importing custom data from third party tools.	Technologies	Y	Y	Y	Y
The product must have a Data API that follows a common standard (such as OpenTracing, OpenCensus or Prometheus)	Technologies	Y	Y	Y	Y

The product must have an installable native application.	Technologies	N	N	N	N
The product's agents must be loadable asynchronously.	Technologies	N [7]	N [7]	N	N
The product must be able to follow Single Page Applications, their internal events and their performance.	Technologies	Y	Y	Y	Y
The product must support JavaScript source maps.	Technologies	Y	Y	Y	N [8]
The product must have live support that is available immediately during the office hours of Veikkaus.	Support	Y	Y	Y	N [9]

Footnotes:

[1] Datadog APM has a system for user and role permissions, but they appear to only control a user's ability to access Datadog's own functionalities. There appears to be no ability to control whether or not a user is able to access data from specific systems being monitored.

[2] Dynatrace has a custom API for personal data erasure. Other providers do not appear to have such a tool, but each of them has a manual process for dealing with them.

[3] Though this feature was requested internally within Veikkaus, no APM tool currently on the market is able to satisfy this. During a call with Dynatrace's representatives, they promised they would have the ability to display such physical devices in their Smartscape map within a few months' time, but it's unknown how this feature is meant to work and when exactly it'll be implemented.

[4] Dynatrace, AppDynamics and New Relic had a "central" location to update their agents from, essentially with the press of a button. For Datadog APM, there is a simple command line for updating, but it has to be run for every instance of their agent.

[5] Dynatrace's "OneAgent" model is one of the features they advertise the most: a one-size fits all solution that automatically detects what any given container or host is running and auto-instruments itself. When a new container is deployed, for instance, it likewise is automatically detected and auto-instrumented. For other APMs, the process appears to be more manual—the correct agent type for the host has to be selected and explicitly installed.

[6] For Datadog APM, any given agent can be disabled with command line, but unlike other APMs, there is no single centralised on/off switch to disable all data gathering.

[7] No APM explicitly supports asynchronous loading. For Dynatrace and AppDynamics, the option exists if configured in such a manner, but it decreases the amount of information they can actually gather, and as such is not recommended and is not particularly well supported.

[8] Datadog APM has no current support for source maps, but a Datadog representative commented that plans to support it in the near future exist.

[9] Dynatrace, AppDynamics and New Relic all had 24/7 support as an option. Datadog APM does not, and their support works strictly in New York time: between 17:00 – 2:00 during business days in Finnish time.