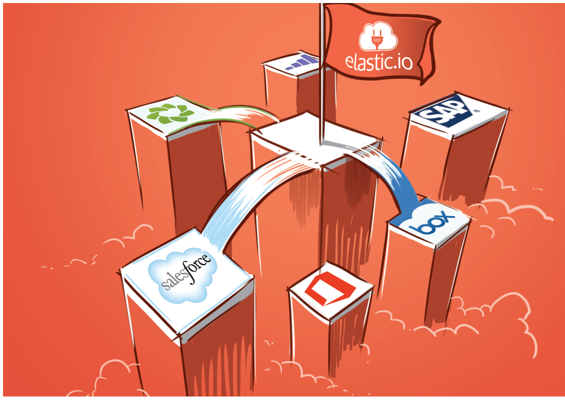

Webinar with Star2Star

How an Inc 500 Company Accelerates Time to Market with iPaaS

Full transcript

Introduction

Daniel Haegele:

00:01 Alright, I think we can start. Welcome everyone to the webinar How an Inc 500 Company Accelerates Time to Market with iPaaS. My name is Daniel Haegele from elastic.io, and I'd like to start with some housekeeping rules: A recording of this webinar will be available, questions can be submitted at any time, there is an extra chat for questions and a Q&A session at the end of the webinar.

If you have any feedback regarding the webinar, it will be greatly appreciated. Just send us a message, or write something in the chat.

Overview of the webinar

01:20 Just to give you a short overview of what we cover in this webinar: First, I will introduce you to today's speakers. Then we'll go on with who Star2Star is, what their solution, the cPaaS, is about, then touch upon the architecture and the use cases. Last but not least we'll give you some more technical information about the elastic.io integration platform. And at the end of the webinar, we'll have a Q&A.

In this webinar you will learn how Star2Star Communications is building its innovative enterprise communication platform, what strategy the company pursuits to address new markets, how Star2Star is using a hybrid integration platform to reduce time to market. We'll cover all these topics, and in addition Star2Star's CTO will also share with you how you can apply his company's experience to deliver your own products and services faster.

Meet the speakers

02:24 The first speaker today is Sergey Galchenko from Star2Star Communications. Sergey has over fifteen years of executive level experience in VoIP technology companies. In the past, Sergey has served as the Director of Technology for Jo-Ann Stores, Inc., CTO at Broadvox, and Manager of Network Design and Planning at Nexbell. Also Sergey has an MBA from Myers University, and a Masters in Computer Science from Don State Technical University.

Our second speaker is the CEO of elastic.io GmbH, Renat Zubairov. He is an experienced speaker at international conferences, user groups, and an active open source community member. During his career Renat was working for big companies like Nokia, Nokia-Siemens Networks, TCS and DHL. Last 5 years he has been working in product start-ups in Application Integration, Data Integration and Business Process Management areas.

Now I'd like to give the word over to our speakers today, and the first speaker today is Sergey Galchenko. You should now have access to the presentation.

Star2Star Communications: Company's profile

Sergey Galchenko:

03:52 Alright, yes, I see the controls. Hello everybody. So, I will start with mandatory Star2Star introduction, who is Star2Star and why we are talking today, why we are meeting today.

So, Star2Star has been in business for almost 10 years now, founded and headquartered in Sarasota, Florida. Beautiful place to visit and to do business in. We are a provider of unified communications as a service. You can think about it as telephony, video, chat, you name it. So, any sort of communication is part of our service.

Star2Star is recognized as Top 10 VOIP Provider by Infonetics. We work with a lot of distributors, and our business model is to deliver our services to a large network of our partners. So, we're currently serving more than 400,000 users across more than 40,000 locations. We have been enjoying exponential growth year to year, especially over the last 5 years.

05:20 In other words, we are recognized by all major VOIP industry-related publications. We've been an Inc 500 for last 4 years, we are in Gartner's Magic Quadrant. And I just wanted to share some representative sample of our customers. As you can see, we work with fairly large retailers. In fact, retailers are one of our sweet spots for our service, and I'll talk about it a little later. We've been in business for quite some time, and we have a lot of great reference customers.

Introduction to cPaaS, Star2Star's product

06:11 Alright. So, now, let's move over to cPaaS. What is cPaaS? It's a term created by Gartner. You know that Gartner loves to create different term, promote them, and unfortunately, they often don't know what that really means. There is really no good definition of what Communication Platform as a Service is.

However, in general, Gartner identifies cPaaS vendors as service providers who have communications-oriented APIs and SDKs at the heart of cPaaS ecosystem. You can think of Twillio, you can think of Tropo; it's a very popular category right now to be in. There are also a lot of traditional phone system providers like Mitel or ShoreTel, who also try to move into that area.

Some of the service providers offer just pure APIs to integrate with their platforms, some are just hosted service or offer pre-configured applications to some degree of configuration. But basically, when you want to integrate your business processes with your telecommunication services, it usually involves fairly heavy development effort to perform that integration using APIs and SDKs.

Star2Star's approach to cPaaS is all about simplicity of use

08:29 Star2Star's vision for cPaaS is a little different than the traditional definition of cPaaS. We definitely would like to offer an API, because we really have to provide it for fairly complicated integration scenarios. We have also been thinking about making integrations more achievable by traditional companies.

If you're not familiar with reactive components, there is a new standard on the web called web components. It basically allows HTML and website designers easily integrate functional pieces of HTML code, which is transparently integrated with APIs. We at Star2Star are planning to add a very simple user interface, almost a drag-and-drop capability, to be able to quickly develop web and mobile applications that will rely on the APIs provided by Star2Star.

Integration should be a significant aspect of cPaaS concept

09:52 Second aspect of cPaaS by Star2Star is integration platform as a service, or iPaaS. Star2Star chose elastic.io as a strategic partner and as an iPaaS provider.

Like in any business, you need to decide what you are good at, and what you are not good at. At Star2Star we are very good at providing high quality voice and video services on a very large scale. But there is also a demand for integrations with all kinds of external applications.

Some of those applications are SaaS, like Google Drive, Office 365, or different CRM systems like Salesforce.com. But there is also a set of applications that are specific to a particular customer. In traditional integration approach, you really need to build a point-to-point integration with a fairly involved and long development cycle.

Integrations with SaaS and custom applications need to be reusable, which is hardly achievable without an integration middleware

11:14 The problem for a service provider is that when integration is done, it needs to be done in such a way that, since I invested time and effort into this integration, I should be able to reuse it for other customers.

We've been trying for quite some time to build connectors to different applications. What we've learned in the process is that it is a fairly involved process, and the results are usually not satisfactory for a large audience. Especially for applications like CRM systems, which are highly configurable: Each customer can have their own business rules around CRM, they can have customized modules, they can create different relationships between custom fields. With all kinds of specifics to a particular deployment, it is very hard to build a generic integration bridge which covers very wide range of possible functionalities.

12:25 At certain moment we started looking around to find out what is available on the market, and we came across elastic.io as a very intriguing prospect to use as integration platform for us. The idea of using elastic.io is to provide a very quick interface to our API ecosystem, as well as a set of communication components for elastic.io, and then let partners and customers perform integrations with third-party systems.

13:13 The last aspect of our cPaaS is Analytics. We are in a fairly unique position to be able to capture pretty much any communications related activity within an enterprise. We feel that by analyzing and restructuring this interaction data, we can provide additional valuable insights to our customers.

Star2Star cPaaS Architecture

13:48 I have a tendency to draw too many connection lines and too many boxes, but on a very high level, the Star2Star cPaaS architecture consists of user interface components which are react components, Star2Star public interface API and Star2Star iPaaS. The backend of all this infrastructure is driven by a set of subsystems: Video, voice, messaging, conferencing - you name it. These are connected through a common message bus. This architecture gives us the ability to very quickly add additional functionalities into our environment.

Different usage scenarios of the Star2Star's cPaaS should be leveraged by its architecture

14:35 We expect that there are going to be different types of consumers of our cPaaS. For example, if we work with developers or with integration partners who perform custom software development, we expect them to utilize a combination of react components and Star2Star APIs, but they can also utilize some functionality provided by iPaaS.

The Enterprise will most likely start using our Star2Star iPaaS in some form or the other, but nothing can stop other applications from utilizing the APIs or react components. Basically, we don't have a single strategy for engagement with a customer. We would like to be pretty accommodating and customer-friendly, depending on our customers' level of comfort with development processes as well as their requirements.

General requirements for iPaaS if to be used as part of cPaaS

15:54 We came up with fairly strict requirements for our iPaaS product offering.

1. First of all, our services are built with enterprise users in mind. Therefore, the iPaaS UI must be easy to understand and to utilize by a power user. If you can create micros or some complicated formulas in Excel, you should be able to feel at home with the iPaaS user interface.

2. Another requirement was easy integration with our Star2Star communication capabilities for future integration with enterprise business processes. The platform needs to provide extended value for the end user, outside of just enabling integration with voice functionality or with chat, - it needs to go beyond that;

3. Another important aspect, which is driven by easy of use, is short cycle from having a concept or an idea to its implementation. Basically, time to value must be very fast;

4. It obviously needs to have enterprise-oriented features like role-based access control, access hierarchy, user identity management;

5. It needs to support different types and styles of integration. You could work with an API and a Webhook, or you can integrate using pre-built components and utilize those pre-built components in the design of business flows;

6. It needs to support complex integrations, obviously. Business processes tend to go fast, and become more and more complicated;

7. As a service provider we really wanted the system to be partitionable, with multi-tenancy, so that we could serve multiple customers.

Developers-oriented requirements for iPaaS if to be used as part of cPaaS

18:40 Another set of requirements was built around developers. After all, we are a software development company, so we would like to make sure that we are using a system that is easy to use for building new integrations. As I said before, it is really hard to build something generic. Usually, you'll need to customize you product offering for different customer's requirements.

You have to support long-running business processes. Your integration flows could sit and wait for certain event to happen, but it also needs to support real-time or near real-time nature of integrations. For example, you would want to have a fairly quick reaction to something that has happened in your chat session.

All these requirements must work together. We went through a thoroughly extensive evaluation of different prospects, and we were very pleased to discover that elastic.io platform meets or even exceeds these requirements.

In addition to that, elastic.io as a company shares the similar vision with Star2Star2 related to how to build scalable, hosted applications, what types of technology to utilize, and so on. So, we found working with elastic.io as a team as well as a platform very easy.

And so, a few months ago we made a decision to integrate elastic.io iPaaS into our service offering.

Use Case for cPaaS / iPaaS #1: Chatbot

20:55 We've been talking about what cPaaS is, why it's awesome. Now I want to give some examples, with which we've been experimenting while we are deploying and integrating with iPaaS as well as building our own set of APIs.

One of the services we are planning on offering is a configurable chatbot. In this particular case, this is hypothetical scenario in which we have a webchat functionality, and the end user is interested to learn about, let's say, the product offer. The agent feels that it's easier to communicate in person via a phone call or a web call, instead of continuing communication in a chat room.

In this case we demonstrate that, first of all, the chat room has the ability to inject structured forms into a conversation. The fact that the user fills out the form with certain information and then submits it, - this action actually initiates an integration flow with elastic.io platform to initiate a phone call between this web user and an agent. This flow will also create a record in CRM that this new user is a prospect, and that there has been a phone call related to this prospect.

Loose application integration

22:55 In this example, we actually use the elastic.io platform for a quick integration between our monitoring application and a chat room. The idea here was to initiate a group chat session, when some certain network element goes into an alarm state. In this case, we extended our monitoring application to support outbound webhooks. We used the elastic.io iPaaS to invite four users into a chat room session as well as allow them to update the status of the element through the chatbot session without leaving the chat room.

In this case our chatbot is trained to recognize certain commands that are typed during the chat session. Upon recognition of a command, the chatbot would submit a webhook back to the iPaaS, which in turn, changes the status of the network element when, for example, we fixed the problem, and the status needs to be changed back to "normal".

In this particular case, we were able to integrate within a matter of hours two applications that don't know of existence of each other, and are not really compatible API-wise. In this case, we were able to do this loose application integration with fairly minimal effort. In the traditional world it would take weeks and weeks to build this particular application integration, find a home for it, test it, and run it. In addition to that, any changes to the code would require quite a complex process of adding extra functionality. Whereas with the elastic.io platform it is quite easy to change the business logic of an integration flow, and so you can very quickly react to changes in business requirements, new ideas, stuff like this.

Use Case for cPaaS / iPaaS #2: Master Data Management

25:50 If you work in Enterprise, then you're probably very familiar with Master Data Management. Typically, data between multiple systems that you have needs to be synchronised. Commonly, this issue is solved using some form of ETL tools that are purpose-built for large-scale ETL integrations.

The problem with the ETL process in general is that it is usually fairly complicated and batch-oriented. You will usually run your synchronisation jobs on some form of an enterprise scheduler, and there will be all kinds of problems related to that. For example, if you haven't designed your ETL job properly, you will have certain data issues, you might fail halfway through your integration, and you will need to go back and figure out what data you synced and what data you didn't sync. It is quite a complicated process, companies have whole teams that are dedicated only to performing ETL jobs.

27:29 We are not different from others. We too have multiple systems that need to be synced. What we decided to do differently, though, is to look at the master data management process at a different angle. We decided to move away from the batch mode, in which you synchronise multiple data records during batch executions, and move towards record-oriented synchronisation process.

With the elastic.io platform we have an opportunity now to detect changes in certain records in our systems of record, and notify integration flows in near real-time fashion about those changes. It gives us an opportunity to have near-real time synchronisation between our systems of record, instead of waiting for batch synchronisation to happen.

28:41 In addition to that, it also gives us an opportunity to have much better quality and visibility of integrations. Even if one of the integrations failed, this means that only one record was affected. And the elastic.io platform also gives us an opportunity to make changes, fix underlying data and re-run failed integrations.

We feel that when we fully migrate from batch to record-oriented synchronisation, we will have much better quality of data in general, as well as much faster synchronisation period.

Use Case for cPaaS / iPaaS #3: CRM Integration

29:38 The last example is quite common in telecommunications world or in cPaaS world - it is the CRM integration. In this particular hypothetical case, which we have been demonstrating to our prospects, the integration flow includes a web form. Here we're showing a Google form, but again, the elastic.io platform allows a very quick integration with any other popular forms.

Let's say you have a Product Inquiry or Contact Me form somewhere on your website. First of all, upon data submission, you would like to have this data to be shown in your CRM. With the elastic.io platform this kind of integration would be a matter of minute to set up.

30:33 The next step would be to check if this new opportunity has certain monetary value that exceeds some hypothetical amount. Now if you remember, on an earlier slide, we have been demonstrating the integration scenario with a chat room. So, you can initiate a discussion about this new opportunity using such a chat room: the data about this opportunity will be then injected into a chat room, and appropriate responsible parties will be invited to this room for a discussion.

In addition to that, we would enable an employee to make a phone call from the CRM user interface to the new opportunity. Then on the background in the CRM, we would create a phone record related to this specific opportunity.

Basically, with this case you would be simplifying your daily routine, which in turn could lead to a much better performance by individual contributors and improve the quality of data collected in the CRM system.

Using iPaaS like elastic.io's makes integrations a much less bureaucratic process

32:10 Gartner has a very good research related to integrations and iPaaS in general. Basically, they believe that in few years at least 50% of enterprises are going to use one or more integration platform as a service in their daily business.

32:47 I've already covered some of the advantages of iPaaS over traditional ETL and SOA related architectures. SOA is another large topic. It calls for existence of Enterprise Service Bus, which is a very complicated exercise to deploy, configure and operate.

33:14 Another interesting side effect of using iPaaS like elastic.io, is that you start thinking about your integrations differently. In traditional enterprise, if you would like to integrate anything with anything, it usually becomes a very long and bureaucratic process. You have to file a request, then do research, locate resources, upon which you might face the fact that these resources are not available. And so, a very simply integration like the one I’ve described before, connecting a monitoring system to a chat room, could become a 4-months project. This is not because it is complicated, but because you have to align resources properly and deal with bureaucracy.

34:15 By using this integration platform, you have much faster time to value, because you can actually work directly with knowledge workers. And they will share with you that their problem is that they, for example, spend two hours every day to do some mundane job, like copying information from one spreadsheet into another spreadsheet, and then they need to email the summary at the end of day somewhere.

These mundane tasks are perfect for utilization of integration platform as a service. It gives very quick time to value, and it also engages people to start using it more for automation of different tasks. You will be able to move from large-scale application to application integrations to user-level integrations for improving their daily cycle.

Typical iPaaS architecture

35:38 In a typical iPaaS architecture, and the elastic.io platform is a very good exampe for this, you’ll have so called components that perform atomic integrations with hosted applications. In addition to that, you will have integration flows, which are basically a set of instructions regarding what to do with incoming data, how to transform it, and what kind of actions to choose with that data.

36:14 There are different styles of integration between iPaaS and applications. Here I’m showing an incoming webhook, that is, of course, if a hosted application is capable of generating this type of message. I also see more and more platforms nowadays offering outbound webhooks, which are basically notifications to remote applications about certain events that are happening within that system.

You can also have a component that periodically checks remote systems for changes in data of particular interest. For example in case of CRM, such data might be new opportunities created within the last 15 minutes.

37:11 In addition to that, you can have an action, which is an outbound event coming from iPaaS. Usually, this action is performed with additional data lookup, or it will create, update, or delete an object within the remote system.

Typically, each of these components produce standardized data delivered to a message bus. Then configured integration flows, which glue together and provide value from these interactions, create business logic and post messages back to the message bus with new targets. In this case this could be an action to initiate a phone call, or to initiate a chat in another system.

This is how a typical iPaaS architecture works.

How you can use iPaaS to speed up time to market

38:23 To summarize, what we've learned in the last few months about iPaaS and how to fit it into your environment: Start small. Find some project that is small but beneficial and will make life easier for certain users. It is important to find those quick wins, because if you're successful in your first few iterations, you will see a snowball effect and increased demand for additional integrations.

39:09 You need to find allies in IT or power users in any other department. There are always people like that, who would like to try new things, who understand business, how it is done in the trenches. What I've learned is that if there is a great idea initiated at the top, it will transform over time, and it will be completely different from the original idea when it is actually implemented at the bottom. So, when you work directly with people in those departments, you deliver exactly what these people expect.

39:55 Another important aspect is education about the software. If people don't understand the platform, they tend not to use it. If they feel that in 5 minutes they still don't understand how the system works, the are not going to use it.

And then you have your normal cycle. After you went through all these exercises, then you deploy the platform, monitor, make changes and repeat on a larger scale.

That's all from me.

Competitive differentiation of the elastic.io iPaaS

Renat Zubairov:

41:51 Hi to all. Well, first, it needs to be said that as a startup, we have all the benefits, but also the drawbacks of a young product that has been launched recently.

Still, one of the significant benefits is focus on the cloud from the moment we started elastic.io. Drawing upon the experience gained in previous companies we had worked for, we started the company with this explicit stress on the cloud-nativeness, multi-tenancy, and web scale.

One of the next differentiations of our product is that for now, elastic.io is indeed the only integration platform as a service that is fully microservices-based. Just to give you a number, currently we run about 800,000 Docker containers per month on our public environment.

42:49 Being cloud-native software, we by definition ensure elasticity and scalability of our product. Sergey has already mentioned high performance and low latency in relation to Star2Star use cases. In addition, these two characteristics are especially important for data synchronisation and complex event processing in IoT use cases, for which our platform can be used as well to connect IoT platforms to cloud-based applications.

Yet another characteristic of the elastic.io platform is its developer-friendliness: Due to the microservices architecture, we can address a wider audience of developers. In addition to that, coming from the IT area ourselves, we feel that we can speak developers' language.

43:33 All this being said, I think one of the primary distinction of the elastic.io platform is focus on APIs. You've heard the word API multiple times through the presentation already. So, as an API-focused integration platform, we obviously integrate APIs, but that's not only it. Just like Star2Star exposes its communication API, we expose our integration API to the outside world.

This makes sense when our platform is utilized for creation of new or better products on top of elastic.io, - products that may not necessarily be associated with data and application integration at all. From our customer projects we have learned that with the end consumer or end user in mind, the best integration is sometime the integration you don't see. You can really compare it with Amazon Web Service: You may be using a SaaS product through your browser, which is hosted on Amazon Web Services, but as an end user, you will never really know this.

45:17 Last but not least, our German heritage is one of our distinctions too, in terms of data security. Because we are Germany-based, we have to comply with the German data protection and data security regulations, which are one of the strictest worldwide. Thanks to that, it is quite safe to say that we, in turn, have one of the highest data protection and regulatory compliance worldwide.

Use Case Wirecard: Onboarding solution for omni-commerce merchants

45:40 In the last couple of minutes I would like to briefly cover a couple of interesting use cases. The first one is about a Germany-based company called Wirecard, which provides an example of API-based integration. Wirecard is essentially a bank, but they more point-of-sale (POS), like a B2B bank. Their customers are primarily eCommerce and offline merchants.

When Wirecard decided to bring its mobile POS onto the market, they found themselves confronted with a similar integration dilemma that Sergey mentioned in relation to Star2Star, namely integration with CRM systems and with customers apps. Quite a large number of different merchants needed to integrate data they already had, for example from Product Information Management Systems, ERP systems, or eCommerce frontend, into the Wirecard's point-of-sale, which is roughly speaking a mobile app.

Obviously, entering such product information is a huge pain, because it is impossible to efficiently enter anything on a mobile device. Wirecard used our platform to seamlessly connect merchants' data sources with their mobile POS deep in the backend, without merchants even noticing the participation of the elastic.io platform in this combination.

Use Case Apora / Aplynk: Self-Service Integration Marketplace

47:35 The next use case is about Apora and Aplynk. This one is interesting because of its multi-tiered value proposition. Apora operates between software vendors and end users, they are the so called middle tier, - what we call cloud orchestrators, or system integrators. They construct their products for particular vertical markets.

Apora is one of such system integrators. They are based in Holland and specialized in implementation of Zoho and Oracle products. Under the Aplynk name, their daughter company, they created a large number of pre-defined integrations that they can deploy and scale easily to a larger number of customers. Also thanks to this latter fact, they can actually offer their integration at a much lower price than compared to custom-built integrations. Which they do through their self-service integration marketplace.

Questions & Answers:

Brian from Cloudera: How does elastic.io compare to Boomi?

Renat Zubairov:

48:57 Well, the main competitive differentiations in terms of technology are the ones that I have already mentioned earlier: These are web-scale, focus on microservices, low latency and high performance, as well as the API-first approach.

Sergey Galchenko:

49:16 So, Brian, I'd like to add that in my previous life, we actually utilized Dell Boomi to integrate SuccessFactors with SAP deployment. What I learned then about them: Their business model is a bit different. They would like to charge you per integration connector, at least it was so a few years ago and maybe changed since then. But back then you needed to tell them upfront what you wanted, and you had to basically buy those connectors. Which could be a fairly large expense on its own. It was tens of thousands of dollars per month for just a point-to-point integration between two applications, one hosted in the cloud and one on-premise. In case with elastic.io, we have access to pretty much any connector that is available on the platform, with no extra charges apart from those for the platform itself.

50:29 Another differentiator from my personal perspective is ease of use as a developer. It's fairly hard as a service provider to get engaged with Dell. They are a large company, and they would like you to act as a large company as well. For me, it just wouldn't be scalable to work with Boomi. Again, I found elastic.io very responsive to our needs, they gave me the ability to start my own development very quickly. They publish source codes for integration components, the system works fairly well, and I like this ability to scale up and down based on the demands.

I hope I answered your question.

The End

Daniel Haegele:

51:38 Ok, we are almost at the end. Actually, we are unfortunately even a bit over time, so we unfortunately cannot answer any more questions. If you have any more follow-up questions, or you would like to have a free trial, please contact daniel.haegele@elastic.io

Thank you very much, Sergey and Renat. And I hope you all enjoyed the information that we gave you.