



# Cloud Native Technology

Will Transform Enterprise Application Integration

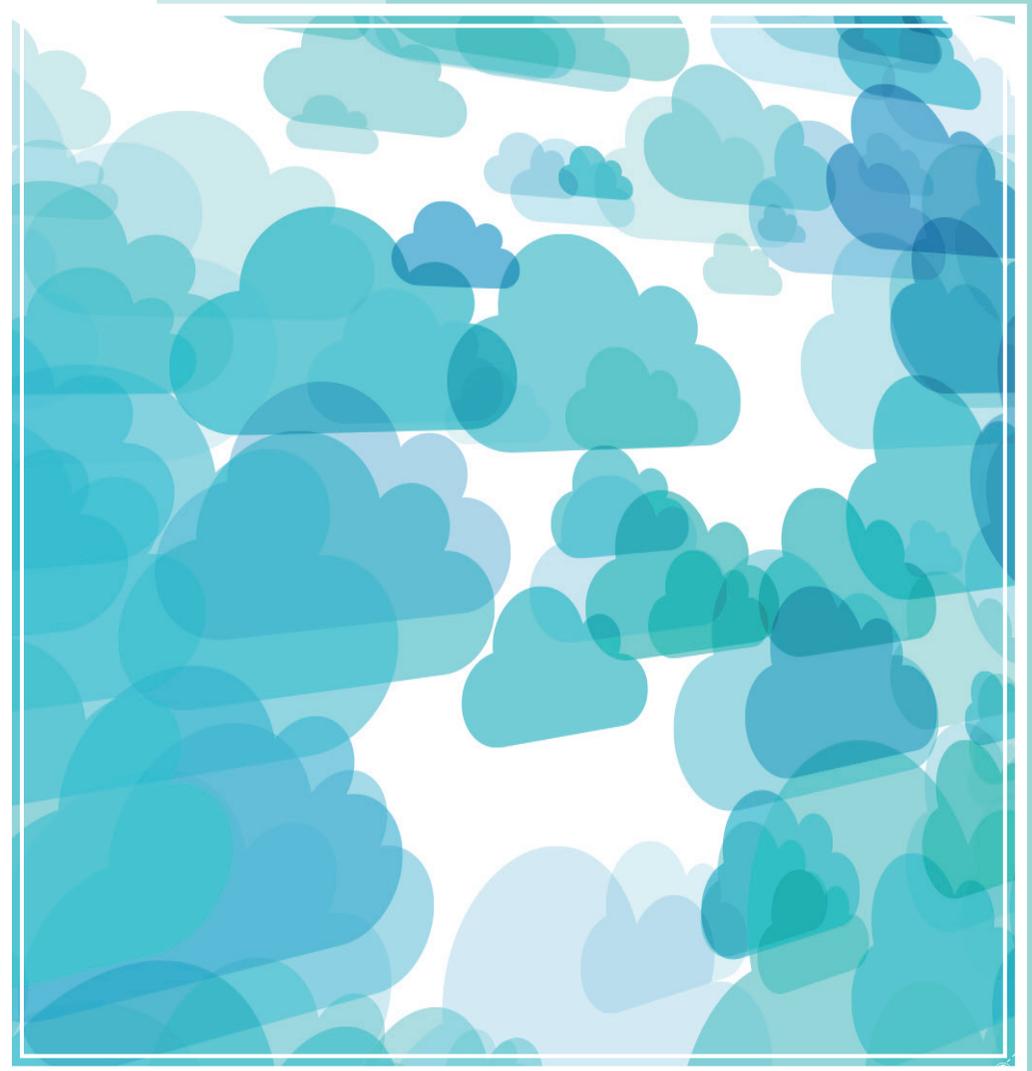


# The Cloud.

At this point it's a hackneyed phrase that could mean anything for how many times businesses have used it. Nevertheless, its growing use reflects the growing importance of cloud technology. Cloud tech has many implications for firms, but one that goes overlooked is the effect on integration architecture, or enterprise application integration: essentially, the melding together of the many applications that a business uses.

The most important part of integration architecture is breaking down data siloes, such that various applications can access all relevant business data, and employees can avoid ignorance of important information simply due to being unable to access it. Enterprise Architecture tools, like Orbus's iServer, provide a similar experience to a well-built integration architecture. iServer's central repository stores data in a variety of formats from many different sources, giving architects visibility over the whole enterprise. Without access to this array of data, the job of an architect would be much more difficult. Full integration architectures need to be even more capable than that, enabling communication across applications to suit a huge number of different functions.

This paper will look at cloud applications in general, but before we proceed it is worth pointing out that cloud native has a particular meaning, and not every cloud application will be cloud native.



“

### **The Cloud Native Computing Foundation provides an Official Definition:**

*Cloud-native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.*

”

“

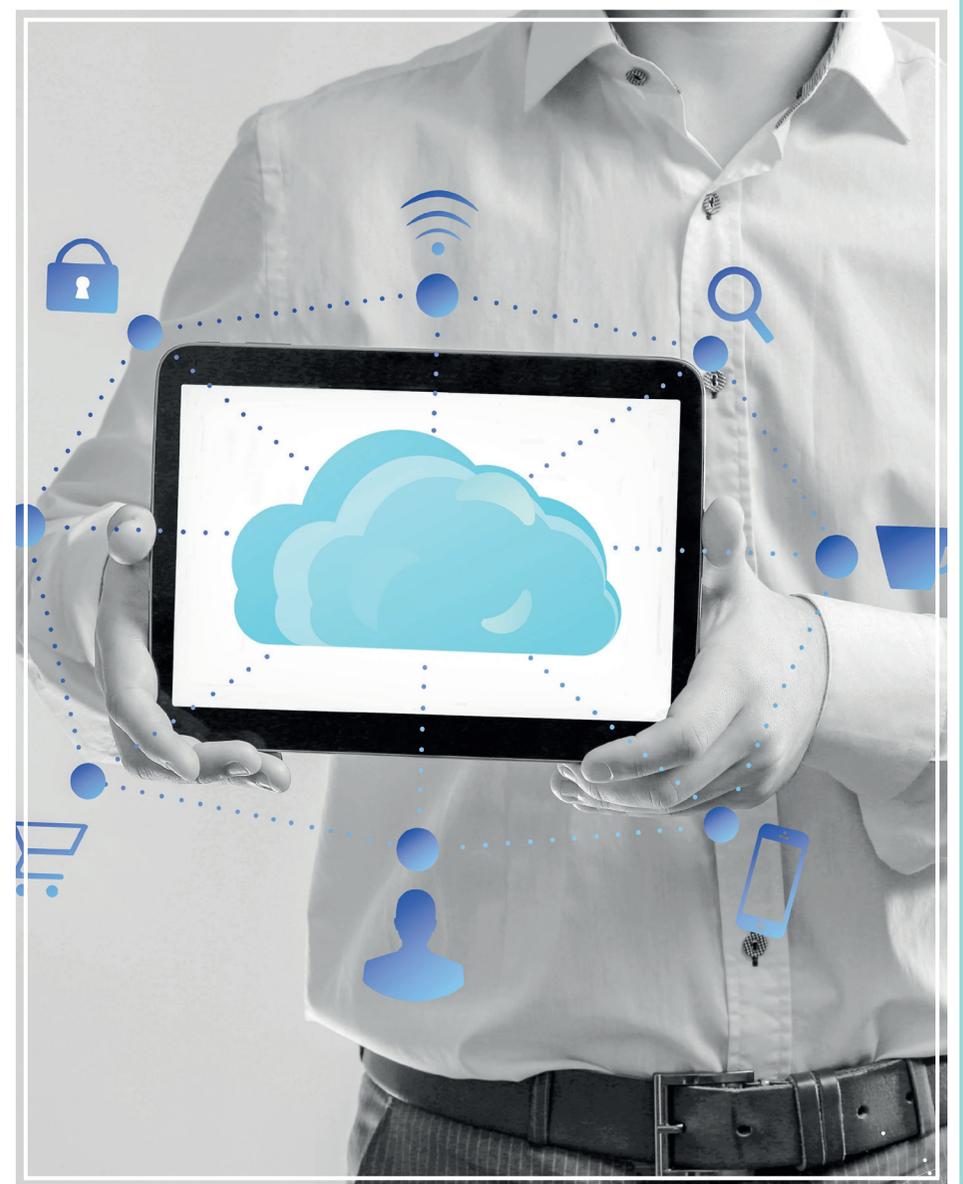
### **Cloud Native applications utilize microservices to deliver value. According to IBM:**

*A microservice is a small application with a small footprint that performs a specific function.*

”

These “microservices” can be deployed, upgraded, improved, and automated independent of any other microservice. This enables frequent updates that does not disrupt the user experience.

The cloud native approach to development has also given rise to a new approach to integration: Agile Integration. As applications are broken down into microservices, the approach to integration can become a core part of their development.





# Cloud and Cloud Native Applications.

Gartner predicts that SaaS products, and cloud computing products in general, will continue to grow rapidly through to 2021, predicting total revenues of \$113B for SaaS and \$278B for the cloud market, compared to 2017 figures of \$59B and \$145B respectively – both almost doubling in 4 years. Meanwhile, IBM suggest 75% of non-cloud apps will be moved to the cloud within the next 3 years.

As for Cloud Native development, the demand for innovation, responsiveness and zero downtime from consumers means that technology will have to adapt towards a more scalable, agile solution.

## Forrester Research predicts:

*In the next five years, the latest apps ... will be available everywhere — in all major public clouds, back on-premises, and even at edge locations.*





# Data Complications.

With on-premise applications, even if data is siloed it is still easily accessible for the business, through databases and files being stored locally. Building an integration or using middleware is not highly complicated in this situation. However, as applications move to the cloud, so too does their data, which renders some forms of integration architecture unviable. The common Enterprise Service Bus (ESB) model of integration has already fallen out of favor due to its rigidity, and the move to the cloud is just another nail in its coffin.

For cloud native development, the microservices approach does promise simpler integration. Nonetheless, Forrester suggests that “distributed application scenarios, such as IoT and edge containers, generate massive volumes of monitoring data and demand a new approach to data collection, analysis, and security”. Much larger volumes of data are sure to introduce additional complexity, even if the microservices themselves are comparatively simple.





# Potential Solutions.

We briefly touched on agile integration above, but its time now to delve into the concept more fully as a potential solution. There doesn't really exist an agreed upon definition for what exactly agile integration is, but IBM offer the following: agile integration—a container-based, decentralized, microservices-aligned approach for integrating solutions that delivers the agility, scalability and resilience required by digital transformation. IBM suggest that agile integration rests on three pillars: Decentralized integration ownership, Fine-grained integration deployment, and Cloud-native integration infrastructure.

Essentially, the cloud native approach to development can also be used in conjunction with agile integration, delivering applications that can seamlessly communicate, take advantage of back-end capabilities and enable innovation.

The core issue with agile integration is how development heavy it is. The Cloud Native approach to development has arisen in response to consumer demand for constant innovation and responsiveness. This means that firms can be expected to deploy multiple times a day - and even with very small deployments, the testing requirements can swiftly add up. That's before having maintain integrations between microservices.





Decentralized ownership rests on having a number of application teams, not to mention the resources necessary to support cloud native infrastructure and rapid deployment – which would likely mean an application team working in parallel with an infrastructure team. Agile Integration is very much a tech company concept, highly suited to organizations with thousands of engineers but quite beyond say, a manufacturing firm with a single development team.

For those in the middle, having plenty of engineers isn't necessarily enough either. Agile Integration is a complicated process, and you can't rely on individuals to be able to master it, which means a skilled project manager or Scrum master is essential. There is also the assumption that every application that will be integrated fits into the cloud native, microservice model, which again won't be the case for any organization which relies more on third party applications rather than inhouse development.

For the more typical organization, they will need a solution that is feasible. As business applications move onto the cloud, so too can integration applications. A major growth segment in cloud applications is in iPaaS, integration platform as a service. iPaaS offerings enable applications, systems and technologies to connect within the cloud and on-premise, offering a solution to firms that choose to go cloud-native. According to Gartner, iPaaS has been one of the fastest-growing software segments since they began tracking it.

A good iPaaS solution offers the benefits of traditional integration architecture, removing data siloes and enabling easier communication between applications. At the same time, an iPaaS solution can deliver faster time to value in comparison to other integrations, and have thus far offered lower costs to implement. Where iPaaS struggles mostly comes from the immaturity of the software; the likes of Microsoft's Azure Integration Services or Oracle's Integration Cloud Service have only been available since 2015, while many of the other vendors are still small firms with limited global coverage and support.

A third party iPaaS offers plenty of benefits, but may not be the solution for every enterprise. While agile integration might be unrealistic, that is not to say that companies can't find a mid point between agile integration and the traditional option of enterprise service buses. Internal development of cloud-based integration remains possible, but comes with additional considerations. First and foremost are the APIs available to you. While every cloud application worth its salt will give developers access to an API, not every API will give you everything you need to comfortably communicate between programs – and even where they do, the amount of development work necessary may be prohibitive. Or, documentation may not be adequate, even if the functionality exists.

This approach is more resource heavy compared to an 'off the shelf' iPaaS or other middleware solution, requiring large investments of staff time, but is most likely within the reach of a well-resourced team. The likes of Kubernetes are both relatively new and complex, meaning there is still limited experience and it takes time to train, but given the importance of effective integration architecture, it is perhaps worth the investment.





## Summary.

For all of these solutions, there needs to be an overarching philosophy and strategy to guide the organization. This is where the enterprise architect and iServer can work in tandem to provide the organization significant value. iServer can be used as a central source of truth enabling engineers to document their application integrations at a higher velocity and prevents disparate data flooding the organization – through iServer and the Enterprise Architect working in tandem the organization will have an increased oversight and modelling capability that will lead to the organization to undertake integration changes at a higher velocity.

iPaaS platforms offer one approach, providing many of the same benefits as on-premise middleware but with a more modern, cloud based design. Agile Integration would be the dream for any IT manager, but is probably out of reach until software development catches up to the fastest moving developments. A middle way that attempts to fuse the centralized ESB style with the APIs for cloud applications could provide a suitable alternative. No matter how your organization proceeds, you'll want to achieve readiness to align IT architecture to the overall digital strategy through visualization and analysis of the integration architecture.





© Copyright 2020 Orbus Software. All rights reserved.

No part of this publication may be reproduced, resold, stored in a retrieval system, or distributed in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the copyright owner.

Such requests for permission or any other comments relating to the material contained in this document may be submitted to: [marketing@orbussoftware.com](mailto:marketing@orbussoftware.com)

**Orbus Software**

Floor 4, 60 Buckingham Palace Road, SW1W 0RR

+44 (0) 20 3824 2907

[enquiries@orbussoftware.com](mailto:enquiries@orbussoftware.com)

[www.orbussoftware.com](http://www.orbussoftware.com)