

Enterprise Application
Infrastructure:
The SOA Gateway &
CaptainCasa
The Ideal Combination
for your Enterprise
Computing Needs



Summary

Enterprise computing used to be delivered using dumb terminals which, while not function rich, provided [OnLine Transaction Processing \(OLTP\)](#) levels of service (Remember them!). In usage terms, OLTP delivered sub second response times thus ensuring that data could be entered and retrieved in real time.

Since the advent of the Personal Computer and laterally the Web browser, there is a constant improvement in the client interface, however, this has resulted in severe degradation, in many cases, of response times. The user experience is adversely impacted through screens that take an age to appear. This is due primarily to two different reasons. The first is that the amount of data to be delivered to the GUI for display can take seconds or longer to be delivered over relatively slow links. The second is that the functionality included in the GUI is so complex, existing Web 2.0, Ajax or other frameworks can take seconds to complete the image.

The [SOA Gateway](#) and [CaptainCasa](#) provide an Enterprise Application Infrastructure using proven Service Oriented Architecture standards to solve both of these issues and to enable organizations to return to sub second response times without compromising on the functionality in the GUI.



About the Author

John Power is a driving force behind the development of the [SOA Gateway](#) and Managing Director of Risaris Limited. With over 25 years experience in the software industry, John has delivered complex integration projects in Software AG, Delta Airlines, Boston University and Morgan Stanley.

Contents

1.	Introduction	4
2.	The Business Case	6
3.	The Return on Investment	7
4.	The Key Challenges	8
5.	Addressing the Challenges	8
6.	The Solution	9
7.	Conclusions	13
8.	About The SOA Gateway	14

V.1.122/01/09

1. Introduction

In the earlier years of online computing, huge effort was put into reducing the amount of data being sent backwards and forwards between display devices and their back end servers. This was done as the speeds at which the devices were connected was extremely slow and the bandwidth available on each of these connections had to be shared with multiple devices. IBM recognised and addressed this many years ago with their 3270 protocol.

While not pretty by any means and restrictive in means of availability of friendly user interface components the 3270 protocol was very efficient from communication perspective: the amount of data for each client request/response was limited by the 80x24 character screens that were sent from the server to the client. This was in particular completely independent from the application processing on the server side – whatever happened there, the communication to the frontend would never exceed the 80x24 character data volume per roundtrip.

With the advent of the Internet, speeds have increased exponentially so the same rules don't apply....or do they? Unfortunately the type and amount of data to be sent backwards and forwards to display devices such as Browsers today has also increased exponentially. As such, organizations today are sending very large amounts of data up and down relatively larger pipes, however, the same issue still applies; if the increase in data is commensurate with the increase in capacity, the same issue arises.

This is most noticeable in countries where broadband speed and availability is relatively slow compared to the rest of the world but also noticeable on the Internet. Consider that the speed at which the data is delivered is subject to the slowest link on the Net-work being used and we're down to the lowest common denominator. So payload sizes *do* matter.

Apart from data volume perspective and from client speed perspective, more and more people realize that the HTML-technology (even though enhanced with JavaScript elements) does not meet the user requirements for heavily used

applications: the rendering speed of HTML/JavaScript is slow compared to more native technologies, the availability of power-user-features (hot keys, fully keyboard driven, ...) is missing, the possibility to integrate sub-devices (scanners, ...) is difficult. - HTML/JavaScript based user interfaces are very dominant in the area of casual usage & anonymous user scenarios because of their zero-installation approach. But they have problems meeting the requirements of heavy-usage, operational scenarios.

CaptainCasa and the SOA Gateway provide an ideal infrastructure to address these problems head on.

2. The Business Case

Providing an Enterprise Application Infrastructure using this toolset provides many different benefits to the business:

- User who are heavily using applications daily, so called 'power users', will be provided with a fully featured Graphical User Interface (GUI) for these applications. This will provide all the benefits of simplicity that a good GUI can bring.
- Consistent response times of less than one second will enable these users to be far more productive. In fact various studies and reports, many going back to the early 80s, have shown that as response times increase, total transaction volume decreases and errors increase.
- Users satisfaction will improve as they can accurately predict how long it will take to complete their tasks due to the consistency in the response times.
- The solution works in tandem with existing systems thus users on legacy systems will see the same data as users on the new GUI system. This facilitates a slow conversion for all users to the new system or both can continue to work in tandem as the business requires.
- Users, particularly in remote locations, will see a level of service they can only have dreamed of.
- Home workers working remotely can also simply use a standard Internet connection to the office and see similar response levels to their office based colleagues who may be running on a network many times faster than their home network.
- The tools provided avoids the big budget integration spends that have been a factor in such efforts in the past.
- The solution can be implemented in small steps thus avoiding the big bang approach and the requirement to commit large amounts of money and resources up front. This means that benefits can be show in a relatively short period of time.

3. The Return on Investment

A major factor prior to the start of any project today is the Return on Investment (ROI) and how it can be measured. The Enterprise Application Infrastructure described provides a number of tangible points where the ROI can be measured:

- Each business transaction will normally consist of one or more online IT interactions. Thus the number of business transactions completed in a day can be measured prior to the introduction and following the introduction of the new infrastructure. As there is a value to each business transaction, it is possible on a daily, weekly or monthly basis to determine what the value of the change is to the business due to an increased number of transactions.
- A well designed GUI interface can drastically cut the costs involved in training new operators of the software, thus new employees can be up, running and transacting business for an organization more quickly.
- Multiple language support in the GUI ensures that the application is written once and a language file translated to provide as many different languages as are required by an organization. This leads to savings in implementation in the first place but also in the fact that a single application can service employees operating in multiple languages.
- Many organizations in the past have upgraded network connections in order to try to improve the response times to their users. Implementing this Enterprise Application Infrastructure can potentially save such upgrades, which can be expensive, or may even facilitate the removal of a previous upgrade to save on costs.

There are also a number of intangible returns which, while not always measurable directly, will certainly have an impact on the business.

- Employees empowered to do their jobs well with these technologies will generally be happier employees thus reducing the rates of staff turnover. This leads to better trained staff, improved morale and so on.
- Customers of the organization will be able to transact business with an organization's operators more quickly due to the quicker and more consistent response times. This leads to happier customers and happier customers tend to return to do more business.

4. The Key Challenges

While it is clear that all of the above will be incredibly beneficial to the business, there are still many challenges:

- In any existing organization, online applications will already exist to run the business from day to day with existing GUI or non GUI style interfaces.
- The likelihood is that these online applications will continue to support the business for some time to come as it is rarely possible to go with a 'big bang' approach. Therefore parallel running for some period of time will be required.
- The new GUI must be implemented seamlessly with the existing systems so that any transaction executed using the new GUI will also be reflected to the older online interface and vice versa.

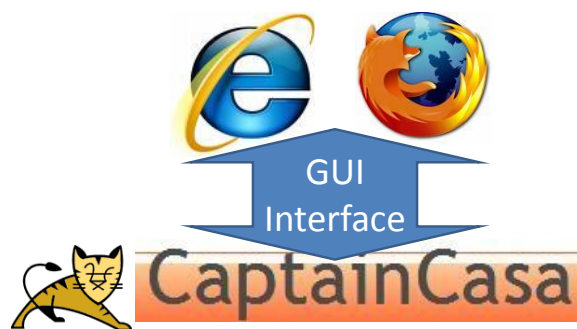
5. Addressing the Challenges

The reuse of existing data and business logic is the key to successfully implementing a new GUI for an application:

- The new GUI implementation must interface with the existing data so that the data upon which the business is built is available to any existing systems as well.
- It must reuse existing business logic as duplication of this logic in different places is likely to lead to different results depending on which online system is used.
- The existing system and new online implementation must continue to run in parallel until it is clear that both systems produce the same end result.
- Clean interfaces between the new and existing systems will lead to flexibility to change the interface as required and resilience when they are changed.
- Existing staff can generally move seamlessly to the new GUI solution due to their familiarity with the older online interfaces that were used.

6. The Solution

The first part of the Enterprise Application Infrastructure solution, and the key to the massive improvements that can be brought to the operation of the GUI is represented by the following architecture diagram.



The user will start their application in their favourite browser as before and the CaptainCasa run time, running in a Tomcat or other compatible server, will deliver the initial image. The key differentiator from this point in is that as changes are made in the browser, only the items that have changes are transmitted via the GUI interface to CaptainCasa. CaptainCasa then builds the output to the browser but again only sends the changes to the browser which has the following impact:

- The payload size for each input and output operation is approximately 1-2K with this architecture compared to "much more data" with a similar solution using HTML for example.
- Intelligence in the browser ensures that data is only transmitted when this makes sense. In other words, data is not transmitted backwards and forwards each time the mouse is clicked on the browser GUI image.

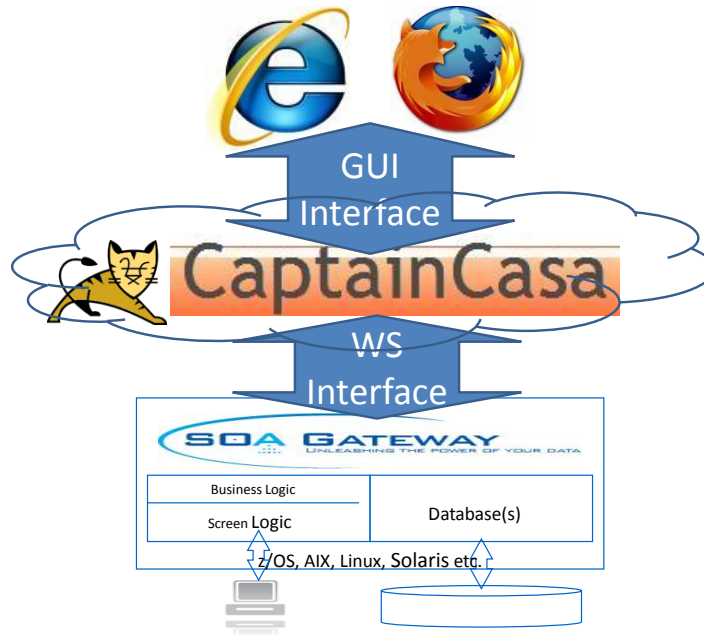
The second part of the Enterprise Application Infrastructure is the novel way that applications and data are made available to the CaptainCasa run time as per the following architecture diagram:



Using the SOA Gateway, the organization's existing business logic and data is made available to the CaptainCasa run time using standards based Web Services. This has the following benefits:

- Existing business logic in COBOL, Natural, CICS and so on can be wrapped and made available as a Web Service in minutes using the SOA Gateway tools.
- Existing data in databases like DB2, ADABAS, Oracle, Sybase etc. can be also wrapped and made available as a Web Service in minutes using the SOA Gateway tools. This is a Create Read Update Delete (CRUD) based service so data may be updated and accessed using these services.
- The CaptainCasa development environment can import the Web Services Description Language (WSDL) from these SOA Gateway services and create a default GUI for these new services.
- The developer may then use the CaptainCasa tooling to modify and beautify the screen layout, include configuration to ensure valid data is entered at the GUI and to deploy to the CaptainCasa run time server.
- The GUI may then immediately be used to start driving the application.
- An additional key advantage to this architecture is that the CaptainCasa run time will only drive the back office Web Service when the input data has been validated against the rules configured using the CaptainCasa development tools.

The architecture allows for a number of interesting deployment options as follows.



In the above configuration, the CaptainCasa server can run within a Cloud instance thus taking advantage of the elasticity that Cloud computing provides for peak loads on the server.



The existing data and business logic could be migrated to a Cloud instance thus enabling the backend and front end systems to take the benefit of Cloud computing.

7. Conclusions

- The Enterprise Application Infrastructure described in this document provides a capability to have rich GUI applications with the benefits of sub second, OnLine Transaction Processing levels of service unavailable with other infrastructures.
- This can be done even though some users will have relatively slow links to the server machines.
- The tooling offered by the infrastructure can ensure that results and benefits can be seen in weeks instead of months or years.
- The infrastructure can work alongside existing deployments, be they GUI or green screen, for as long as is necessary from a business perspective.
- All of this can be achieved with minimal initial outlay based on the usage based model adopted by the infrastructure.

8. About The SOA Gateway

The SOA Gateway is a cost effective software tool to:

Access data faster...

It enables access to data from a wide range of database languages (ADABAS, MySQL, DB2, VSAM, Oracle etc.) without server side code, or expensive middleware.

Access business logic easier...

The SOA Gateway enables easy access and re-use of valuable business logic available in CICS, COBOL, C, NATURAL and many other languages and environments.

The [SOA Gateway](#) is developed by [integration specialists Risaris](#) Limited.

For a free trial of the [SOA Gateway](#), please visit:

http://www.soagateway.com/html/registration_form.php

This document is distributed for information purposes only and does not form part of or constitute an agreement with Risaris Ltd. Although Risaris Ltd. uses reasonable efforts to include accurate and up-to-date information in this document, Risaris makes no warranties or representations as to its accuracy. Risaris Ltd. may also make improvements and/or changes to this document at any time without notice. The various approaches outlined in this document are put forward in good faith, but it remains possible that individual results may vary. For that reason and in accordance with standard practice, readers are encouraged to test any materials developed on the basis of this paper before putting them into productive use.

More whitepapers at:

http://www.soagateway.com/html/industry_papers.html

See Terms & conditions at www.soagateway.com © Risaris Limited 2009.