abavus Transforming Organisations

White paper

Designing effective system integrations for UK local government

Contents

For further information about implementation and resourcing please contact us.

abavus
iTouchVision
myCouncilservices

Visit www.abavus.co.uk
Email info@abavus.co.uk
Call 020 8530 2505
Tweet @AbavusLtd

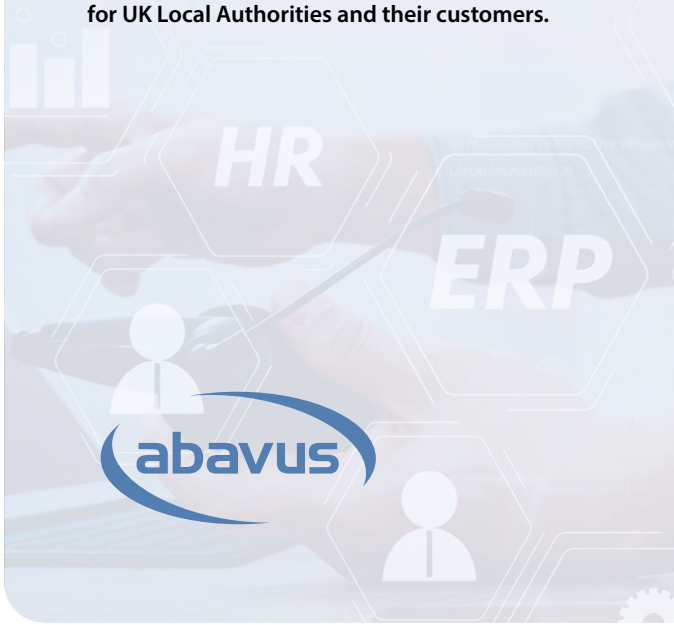
The challenges and opportunities of system integration for UK local authorities

This white paper will explore and explain the lessons that we have learned over many years of delivering system integrations for our UK Local Authority clients.

The delivery of secure, system to system integrations has become a staple delivery of most implementation projects. System to system integrations are important elements of a project because, when done well, they enable the Local Authority to create the maximum number of opportunities for process automation and efficiency gains.

Abavus has always taken an open approach to integrating third party systems. When we say open, we mean that in two ways:

- **We fully support contemporary open technical standards for system to system integration.**
- **Organisationally, Abavus is open to integration with any third-party system and collaboration with third-party suppliers in the pursuit of an improved outcome for UK Local Authorities and their customers.**



The My Council Services platform has been engineered to integrate with any third-party solution enabling seamless and effective end-to-end digital processes. This is enabled through a comprehensive API made up of a full set of fully documented web services.

The opportunities for effective integrations are the ability to create and deploy truly digital processes, from start to conclusion. Processes that include a system to system integration can be automated to the maximum degree that is appropriate for that process. When implemented well, integrations will save time, deliver cashable savings for the Council and enhance the user experience.

Delivering an integration is not without its risks and challenges of course. Any integration will have its own degree of complexity and technical details. The initial delivery can be time consuming and labour intensive. In that context, any Local Authority considering a technical integration between different systems should think and plan carefully before embarking on a project.

Abavus has now completed hundreds of individual integration projects from the relatively simple payment integration through to the most complex multiple flow integration involving sensitive financial and personally identifiable data. This white paper shares our accumulated experience and best practice.

Overview of our work with UK local authorities

Abavus's activities focus exclusively on UK local authorities, enabling each organisation we engage with to achieve a wide-ranging transformation of service delivery across the full range of Council functions. We work closely with our clients and have invested significantly in the development of the My Council Services digital platform and its Application Programme Interface (API) to support processes that require the creation of integrations between a wide range of both on premise software systems and cloud platform.

The UK Local Authority sector has a long history of investment in technology and associated systems. It is surprising how many different vendors operate in the space and the proliferation of different systems and applications (both on-premise and now increasingly cloud-based) that are inactive use.

The schematic to the right shows a small selection of the range of systems and applications to which Abavus has delivered integrations of one type or another.

Types of integration

In this white paper, we introduce the main types of integrations.

Typically, there are three primary types of integrations used within local government:-

- **A synchronisation**
- **A lookup**
- **An output**

All three of these characteristics can be present in a single integration (which of course will add to the overall complexity). Before we progress into the detail and best practice methodology it may be useful to recap on the definition and purpose of the three classifications of Local Authority integrations listed above.

Synchronisation

Most Local Authority systems will have the concept of a case, a worksheet, a job ticket or a service request. This is generally an issue or something that needs to be dealt with. This may be resolved by the Local Authority's own staff or by an outsourced contractor. You can refer to any of these items i.e. a case, a job ticket or a service request as a 'container entity'. Each of these container entities will hold information that is relevant to the service request or case such as the submitters details, location information, notes and status flags. The relevant and required data fields included in each container entity will need to be securely shared with and accurately mapped to the corresponding fields in the recipient database system. Synchronisation integrations help improve the Council's data management and to create a consistent and federated data infrastructure.

Lookups

It is not unusual for UK Local Authorities to hold data in separate databases that are not connected with the wider information architecture of the organisation. These can be thought of as data silos. This could be a spreadsheet (although we would advise against using spreadsheets as long-term data storage solutions – just ask Dido Harding), an access database, SQL Server database, or any other 'flavour' of database technology. Just because the data held in these isolated locations and is separate from the wider information management set up, does not mean that it is not useful and valuable. It often includes address records, financial data, information about an asset or personal data that needs to be referenced for other processes. To aid a process sometimes you will need to validate data provided or retrieve data to enhance or verify the process. For example, cross checking via a lookup to birth and death registrations that an applicant is of an appropriate age to apply for a benefit. Lookups will normally be a real-time integration happening at the same time a customer enters information and waits until a response is returned. Lookups can make digital self-service processes more efficient and easier for a customer to navigate, by reducing the need to re-enter information that the Council already holds about a person for example, they can be extremely useful and often make the difference between a really good user experience versus a sub optimal one.

Output

The most commonly occurring example of an integration that would be classified as 'output' is a payment integration i.e. the software systems that Councils utilise to collect online payment such as CivicaPay, Worldpay or Capita eStore. There are other examples of output type integrations, for example, when integrating to a document generation system, or notification process such as email or SMS. As the name suggests the object of this type of integration is to produce and output that typically concludes a process or at least concludes on element of a process (the successful completion of a payment, the production of an official document, badge or permit or the send and delivery of an electronic communication).

Best practice methodology for integrations and design considerations

Understand the business process

One of the first things to be done, when designing an integration, is to ensure that there is a documented understanding of what the customer is trying to achieve as an outcome for the integration and more broadly what the process needs to enable. This is best done through a business mapping process. This is time-consuming and it will likely require the input of multiple stakeholders (depending on the complexity of the process). Done properly this process mapping exercise is an insurance policy against sub-optimal outcomes further down the line. Process mapping will enable proper feasibility versus investment assessment, and it will provide an opportunity to streamline processes.

Once the mapping of the business process has been completed, it is then possible to overlay the specific integration points. When re-engineering any solution it is essential that people (service users and service administrators) are considered first, secondly the process itself, and finally the system. Failure to design it in this way may mean the business process does not support the people who use the service and the colleagues who administer and support the process. This can result in a poor user experience and unnecessary and unhelpful changes to workflow and efficiency.

Information capture

The golden rule for information capture is to ensure that you collect all required information from the service user as early in the process as possible.

For example, when a customer submits a request, they will be asked to provide information to support their request. It is important to validate as much of this information before the customer submits the request. Unvalidated information will cause issues downstream when the integration runs which could involve someone updating the data manually or worse, having to contact the customer for more information. All of this creates a delay in processing the request and uses scarce internal resource time.

Some examples where data can be validated in the My Council Services solution include:

1. **Addresses - The customer has the facility to be able to select their address from a list, which is validated against the Local Land & Property Gazetteer (LLPG). This address lookup ensures the data is integrated from one system to another offering seamless integration.**
2. **Accounts - If you are asking the customer to enter account information such as a benefits number, council tax account number, and job seekers number. It is important that information like this is validated to ensure the correct information has been entered. Mistakes at the time of submitting the request will wreak havoc when the first integration point is reached. Failure to get the correct verifiable information will cause an integration to fail and will require the advisor to follow up with the customer.**
3. **Financial - Validation of bank account numbers or card payment details should be validated on entry to preempt any fraudulent activities and genuine data entry mistakes.**

Capture clean and complete data

It is crucial that before locking down any downstream systems the data captured at the start of the process is both clean and complete. This consideration is a feature of eform planning and design. It is of critical importance that eforms are design to capture all the correct information. When capturing information, it is key to observe the rule that initial data capture scope is constrained to the lowest system specification that exists in any integrated application that is part of the process. What that means is that all data capture must be designed to comply with the lowest common denominator (in terms of specification) of any downstream system. Validation must be applied to reflect that which the lowest common denominator dictates.

To follow are some examples of validation protocols that would need to be determined by the lowest specified system:

1. **If a field is required by a downstream and integrated system, make it mandatory, not allowing the customer to submit the form until that field is completed**
2. **Consider if a downstream system has a maximum field length. If it has, make sure eform fields only allow the customer to enter data to that maximum field size. Trying to push 100 characters into a field that only take 50 characters will cause the integration to fail. Cutting off the remaining 50 characters will mean not all the information is passed to the downstream system and will unquestionably complicate the delivery of the service requested.**
3. **Validate data format and avoid capturing information that gets truncated. If you are asking for a mobile number, make sure you check the length is correct!**
4. **Validate the information entered. If you have a number only field in the downstream system, make sure you only allow the customer to enter numeric values.**
5. **If one system has a maximum file size of 1MB, then the validation needs to be applied to ensure the customer does not upload files above this limit.**

One thing that commonly breaks an integration is the use of emojis as these are new character sets. Many legacy systems cannot support them, and either fail the integration or write the actual character to the field i.e. U+1F60x = 😀.
To complicate things further new character sets are constantly being added all the time. Based on our experience any integration would be much more robust by not allowing the use of emojis. A good recent example that caused an issue was submission of a service request via our mobile application on a smartphone, the user included an emoji in the device name, this was flagged an issue as part of our web services monitoring and required a change to the configuration.

Fault tolerance

With any integration, there will be errors you have not been able to forsee (however carefully you process map) and it is important to design an integration to present a clear message to the system users explaining why the integration has failed and how the issue can be corrected.

An issue that we have encountered and had to build into our best practice catalogue is related to new addresses. This is common because of ongoing residential development in many parts of the country to provide much-needed homes. My Council Services takes daily feeds for the Local Land and Property Gazetteer (LLPG). This enables a customer to select their recently added property as soon as the Council has accepted it. What became apparent is that many older downstream systems that were part of the integration overlay may not have imported the most up to date copy of LLPG. Consequently, when a customer selects that newly added address, the most up to date system attempts to push the address into the downstream system which is running on an out of date LLPG and the address is rejected. This causes a failure.

The issue described above required a workaround to be implemented. In future cases, when LLPG mismatch happens on a new property, a not is added to the service request that explains the nature of the problem, how to resolve it, and change the status to 'Attention required'. This workaround was appropriate in this case because it prevented the integration from trying to process again and failing but also to notify the system users of the nature of the problem.

Once the issue is resolved the system user is able to change the status which means at the next integration cycle the service request will be integrated.

Integration approaches to avoid the common faults of failing

Plan for downtime

If you do not design an integration to take account of downtimes (planned or unplanned) a likely outcome is that your development teams will end up unpicking partially integrated requests. This can be time-consuming and is completely avoidable.

When designing an integration, our best practice tells us to follow these basic principles: -

1. **Run a check to verify if a request has been previously integrated, before attempting to integrate.**
2. **Once a service request has been identified that is still awaiting integration, change the selection criteria e.g. the status to 'in progress', to stop other flows from trying to integrate it again during the current flow.**
3. **Once the identified service request has successfully integrated, save the ID from that system against the service request or entity.**
4. **Change the status of the service request or entity to indicate it has completed.**

If at any stage the server is restarted or there is a loss of network, the integration can pick up where it left off ensuring no duplicates are integrated into the downstream system. Where a failure occurs at stage 2 above, monitoring will need to be in place to ensure no service request or entity has this status for more than 30 mins. The system users will need to reset the status, or an automatic rule could create the status back to an integration triggering status.

Understand the pattern and peaks of demand

With any customer-facing service offered by the Council, there will be peaks during the week, month or year depending on the cyclical nature of demand. During these variations of demand, inbound traffic could range from a few requests per day to thousands per day. A recent example of this, that we have seen, is Garden Waste Collection Services, especially where a Council charges for this service which is increasingly common. Typically, with Green Garden Waste, customers want to sign up as summer approaches and the grass begins to grow and this is reflected in patterns of demand. Similarly, schemes that are available for short periods such as bus cards for school-aged children will have obvious and significant peaks and troughs in demand.

In these situations, where demand peaks and recedes, it is important to understand the maximum load that the downstream, integrated system can process. This constraint should be used to govern the integration flow, creating a drip filter to only a sustainable number of requests at a time. This throttle on the throughput will extend overall processing time slightly but it will ensure that all cases are processed accurately and consistently, allowing the downstream system to accept the new data, avoiding performance issues or worse, processes becoming overwhelmed and grinding to a halt.

Concurrency

It is not at all unusual for a system to system integrations for Local Authorities to require multiple flows. This means that entity containers (service requests, cases, tasks, etc.) travel in multiple directions at the same time. Concurrency is an advantage because it means that your integration is effectively multi-tasking which can be much more efficient, doing more than one thing simultaneously. That said, integrations that create concurrency, need to be carefully analysed and designed Depending on the range and complexity of data being passed as part of the integration and the time taken to process that data at each stage of the flow, it is not inconceivable that two flows are running simultaneously. If the integration is not designed properly, and the metrics of data volumes and processing times are not accurately calibrated, the multiple flows can end up consuming the same data. If this occurs the integration can begin to create duplication problems. Unless your Enterprise Service Bus (ESB) can prevent parallel processing, the integration process will need to govern the cadence of repeat processing.

Cyclical integration

A cyclical integration is an integration in which the upstream system (My Council Services for example) integrates into a downstream system (IDOX or another line of business application for example). The downstream system then processes the update, and then passes information back to the upstream system again, which in turn triggers the upstream system to integrate into the downstream system and so on. As with concurrency described above, it is very easy to create a system integration that is repeating itself. For example, adding the same note to each system many times.

A good example of this is the need to keep two systems and the information they each hold in synchronisation. Each system has a process when say, a note is added, to push this updated information to the other system.

When creating cyclical integrations, we recommend doing this using note types and a note key.

1. **On the upstream system, only integrate NOTE_DOWN types, with the start of the content holding the service request or entity ID. Then push the note to the downstream system.**
2. **When the downstream system accepts the note and creates it locally, it then tries to push this to the upstream system. If the note content has the service request or entity ID, the note is discarded, if it does not, the note is created as NOTE_UP type.**

This method ensures both systems are in sync and duplicated notes are not added.

Keep the customer informed & updated

Nothing is more frustrating to a customer than making a service request and then hearing nothing at all or waiting days to receive any form of response or acknowledgment. These types of situations lead to a failure of demand i.e. failure to do something – turn up, call back, send an acknowledgment message – that in turn causes the customer to make another demand on the system, like phoning up the contact centre to complain.

For each service request that is received from a customer, there is a process that is triggered with both system / technical and human actions to be completed. A customer will not be aware and may not care about any of this and just expects the request to be completed promptly and efficiently.

To effectively manage this when the customer makes a request the first notification should be to inform the customer that you have received the request, and what your service level agreement timescales are for dealing with the request. This should detail how long the process will take, what the required steps to completion are, and when they can expect their request to be resolved. These can all be catered for through automated messages through the channel of the customer choosing e.g. an email, SMS, push notification or an automated call.

Assuming that you have informed the customer of your SLA timescales, they will still be interested in the progress of their request. For the relevant stages of a process, share this with the customer proactively. Stages that are 'in progress' are useful for a customer to see and are important because they demonstrate progress. Well designed and timely communication that is automated, accurate, and triggered as part of the integration workflow will help your Council avoid the perils and wasted resources of demand failure.

Monitor your active integrations

With every integration that is designed and implemented, despite our best efforts, things can and will occasionally go awry. When we deploy integrations for our Local Authority clients, we implement a robust monitoring toolset which keeps a constant and watchful eye on what that specific integration should be doing. The monitoring tool provides real time information and alerts on the frequency of web service methods being called including which integration is calling them, it also counts the historic volumes. From this real-time and retrospective metric, it is possible to see a drop-in calls, peaks and troughs which enables a constant health check on the performance of the integration.

Conclusion

Completing integrations should never be underestimated or embarked upon lightly. Even simple integrations can have hidden complexity and unintended consequences. Abavus have completed hundreds of integrations for our Local Authority clients, right across the range of integration types and into a bewildering array of different systems and platforms. This white paper, I hope, provides a useful checklist. As a minimum what I have shared here may validate what you already know and it may also provide you with you some key learnings for the next time you are planning a system integration. I recognise that what is shared here is not exhaustive. If you would like to discuss any aspect of integrations that you are considering, please do get in touch with me at **mark@abavus.co.uk**.

abavus Transforming Organisations

About Abavus

Abavus is an established UK-based technology and service provider to the UK public sector. Since 2007 we have been delivering software solutions and consultancy services that underpin and enable transformation and efficiency programmes in local authorities.

At Abavus we work extremely closely with one innovative, independent software house – iTouch Vision – who develop the widely used My Council Services enterprise platform. Abavus and iTouch Vision have entered into a long-term partnership agreement, whereby Abavus implements and supports My Council Services exclusively in the UK public sector.

This allows iTouch Vision to invest the maximum resource in its continued research and development activities, whilst Abavus focuses on the delivery and support of the solutions to clients.

Abavus has a customer base of over 50 UK local authorities and government organisations using the My Council Services solutions, as well as a small number of private clients. We are firmly established in the public sector market as a leading provider of innovative, cost-effective information technology solutions, helping public and third sector organisations to save money and make significant efficiency gains.

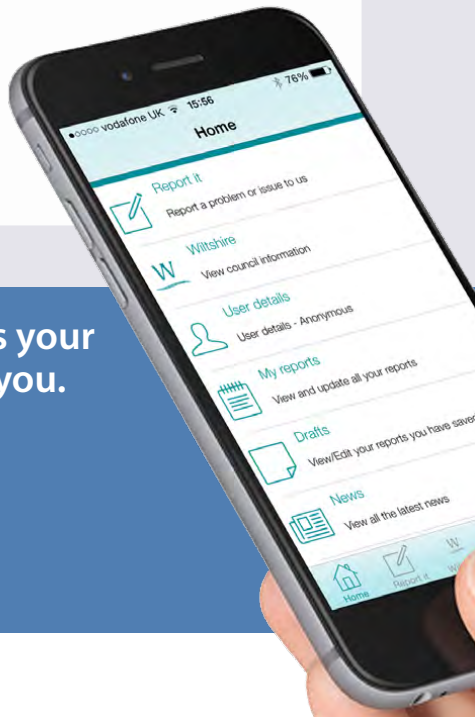
Abavus has been successfully delivering technology-enabled transformation projects in the testing operating environment that has prevailed following the financial crash of 2007 and the unyielding constraint on funding that this has created. Whilst this has been challenging it has meant we have to keep our projects focused on value and delivery. Our responsive, customer-focused approach has enabled us to thrive in a busy marketplace.

Consultancy and training

Abavus also provides consulting and training services to ensure that our clients are up and running with the technology solutions we provide as quickly and efficiently as possible.

Drawing upon our extensive knowledge and understanding of public sector organisations and the specialist technical skills within the team, Abavus leads and supports consultancy and training engagements with the focus on driving business transformation and change.

myCouncilservices



abavus

iTouchVision

Please feel free to contact us to discuss your requirements. We'd love to hear from you.

Visit www.abavus.co.uk
Email info@abavus.co.uk
Call 0208 530 2505
Tweet @AbavusLtd