Using multi-tenancy in Microsoft Dynamics CRM 2013 to address challenges in enterprise business environments

Version 1.0

Author: Roger Gilchrist
Company: Microsoft, Ltd.

Contributors: Bernt Bisgaard Caspersen

Released: September 2013

Applies to: Microsoft Dynamics CRM 2013
Microsoft Dynamics CRM 2011

Microsoft Dynamics CRM

Copyright

Feedback

To send comments or suggestions about this document, please click the following link and type your feedback in the message body: http://go.microsoft.com/fwlink/?LinkID=311837

Important: The subject-line information is used to route your feedback. If you remove or modify the subject line, we may be unable to process your feedback.

Table of Contents

Introduction

There are a number of complexities associated with implementing large-scale CRM projects in enterprise business scenarios. In these situations, using multiple tenants in Microsoft Dynamics CRM 2013 can help to address several typical challenges, which include:

- *Functional localization*, for organizations with different business units or areas that have varying business requirements or processes
- *Master data management*, for organizations that need to distribute but maintain control of business data in certain ways
- *Physical distribution*, for organizations with user base that is physically distributed in ways that introduce challenges such as distance of connections
- *Security and privacy*, for organizations that need complex control of distribution or access to data
- *Scalability*, for organizations with workload requirements that exceed or are prohibitively costly to host within a single system

Microsoft Dynamics CRM offers a variety of capabilities that allow customers to implement rich solutions to address complicated business requirements, which tend to have an even higher degree of complexity in enterprise environments.

When considering how best to address these types of challenges, a key factor is determining whether to:

- Use the multi-tenancy capability that is native to Microsoft Dynamics CRM, or
- Set up completely independent Microsoft Dynamics CRM instances.

For solutions that incorporate multiple Microsoft Dynamics CRM instances, a second major consideration is any potential need for integration or interaction between the different instances in the overall business solution.

Scope

This white paper discusses scenarios in which using multiple Microsoft Dynamics CRM instances to separate areas of functionality can assist in addressing business challenges in the enterprise. The paper also describes some common approaches for integrating multiple, separate instances of Microsoft Dynamics CRM. Included is detail on terminology and information on the benefits and potential limitations associated with each approach.

Important: This white paper is designed to help Architects, Consultants, and Technical Decision Makers understand various high-level approaches for using multi-tenancy to address key challenges in enterprise business environments. However, the paper is not designed to provide prescriptive guidance on implementing any specific solution. For additional information about implementation techniques, in this document, see *Appendix A: Additional resources*.

Applicability

The information provided in this white paper applies to:

- Microsoft Dynamics CRM 2013
- Microsoft Dynamics CRM 2011

Terminology: Instances, organizations, and tenants

Before going into the details of scenarios that might require using multiple instances of Microsoft Dynamics CRM, it is important to understand the associated terminology and some of the options that are provided in Microsoft Dynamics CRM for hosting and segregating functionality and data.

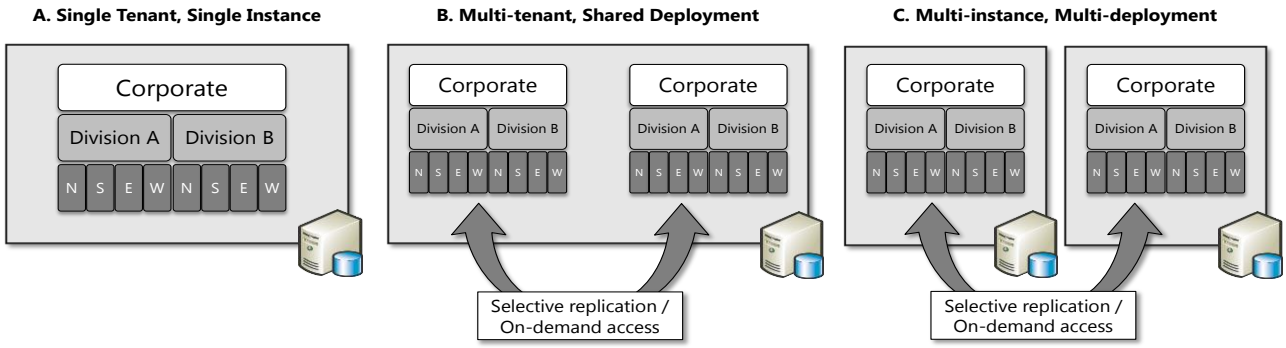
By default, installing Microsoft Dynamics CRM creates a single deployment, or *instance*, of the product. This instance can include a number of servers, such as one or more:

- SQL Server instances hosting:
  - One Microsoft Dynamics CRM Configuration database
  - One or more Microsoft Dynamics CRM Organization databases
- Microsoft Dynamics CRM web servers hosting the web application for the deployment
- Microsoft Dynamics CRM Asynchronous servers hosting asynchronous processes for the deployment
- Microsoft Dynamics CRM Email Routers (optionally)

Collectively, this group of servers constitutes a single Microsoft Dynamics CRM instance.

An *organization* is a logical grouping of metadata and business data that is maintained logically separate from other business data. Microsoft Dynamics CRM also offers a model that provides for sharing the underlying physical resources within a single deployment across multiple, different organizations. To accomplish this, Microsoft Dynamics CRM creates *tenants*, which are technical groupings within the application, such as separate user databases within SQL Server. Though the SQL Server instance and Application servers are shared, there is a separation within Microsoft Dynamics CRM that maintains a logical separation of organizations.

Each organization is maintained within a tenant, and each tenant can support one and only one organization. A Microsoft Dynamics CRM 2013 instance can support one or more tenants, and therefore can host one or more organizations.



Because each tenant or organization is a separate set of configuration metadata and data that runs on a common infrastructure, multiple, smaller organizations can share hardware for scenarios in which supporting each organization with dedicated infrastructure would be impractical. Using a shared infrastructure can, however, pose challenges around isolation from a security and resource usage perspective. For example, an “expensive” report running in one tenant can impact the performance of other tenants that are sharing the infrastructure.

An alternative approach would be to deploy multiple, separate instances of Microsoft Dynamics CRM, each on a dedicated infrastructure. This option offers the benefit of isolating each tenant, but it also requires the administrative overhead associated with installing and maintaining multiple sets of infrastructure and to manage the physical resources. For larger scale deployments, the associated impact may not be a significant factor when compared to the benefits, but this should definitely be a consideration for smaller scale solutions.

This paper focuses on scenarios in which deploying multiple Microsoft Dynamics CRM tenants can provide benefits. In some scenarios, a single instance using multi-tenancy will be most appropriate, while for other scenarios it will be more effective to host multiple tenants, each on a dedicated infrastructure. What is most important is to understand how best to take advantage of two distinct but overlapping approaches:

- *Using separate tenants* to segregate Microsoft Dynamics CRM functionality
- *Using separate instances* to physically or logically separate a Microsoft Dynamics CRM deployment

This paper considers each scenario within the context of addressing business challenges that are common in enterprise environments.

Common business scenarios

Developing solutions that use either separate tenants or separate instances can help to address challenges in a number of commonly occurring business scenarios. A review of these scenarios highlights some of the underlying factors that can lead to a decision to separate a solution into different tenants or instances, which in turn can be useful in helping to identify other situations in which this approach might be beneficial.

Business scenarios that enterprise customers commonly encounter include (but are not limited to):

Scenario	Description
Functional localization	• Provide a common core of functionality, but with local functional variations – with "proper" delegation
Master data management	• Provide a consistent, managed core of information, perhaps distributed only selectively • Enable smooth transitions without big bang replacements
Physical distribution	• Mitigate network latency
Security/privacy	• Accommodate legislative/national differences (e.g. patient confidentiality, Swiss banking, third-party use)
Scalability	• Accommodate extreme volumes and/or extensive use of Service Scheduling • Provide for isolation of workloads (e.g. web site, customer kiosks)

The following sections consider each of these scenarios in greater detail.

Functional localization

Business need

This scenario typically arises in organizations with overlapping but separate functional needs. Some common examples include:

- Organizations with different business divisions, each with a different market or model of operation.
- Global businesses with regional or country models that differ to account for variations in approach, market size, or compliance with legal and regulatory constraints.

In these types of business environments, an organization often will have common sets of functionality that allow specific regions, countries, or business areas with a degree of localization regarding:

- Information capture, for example capturing the ZIP Code in the United States would correlate to capturing the Post Code in the United Kingdom.
- Forms, workflows.

Solution and constraints

Using multi-tenancy or multiple instances allows each business area or local region to have an independent implementation with its own local variations. While this approach does allow greater degree of flexibility, businesses typically want to drive as much consistency as possible to:

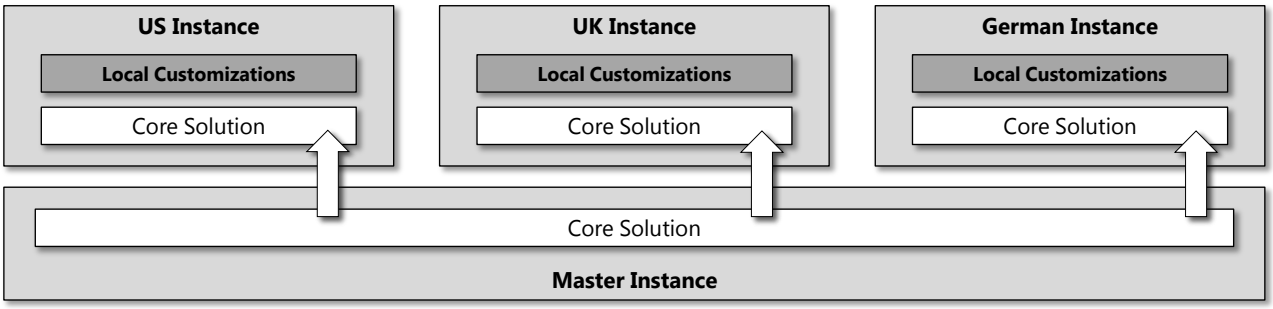
- Maximize reuse of implementation investment.
- Control specific core elements to drive commonality and consistent reporting across all the tenants in the business solution.

A great approach for accomplishing this level of consistency across instances is to use the Solutions feature in Microsoft Dynamics CRM 2013. Simply create a customized Microsoft Dynamics CRM Solution for each region, and then import each customized solution into a separate "regional" organization.

Note that there are limitations on using Microsoft Dynamics CRM Solutions in these types of scenarios. For example, consider the inability to include reference data in a Microsoft Dynamics CRM Solution and the lack of granularity of

some configuration overrides. This limits the ability to offer fully localized variations using Microsoft Dynamics CRM Solution, but it can be extremely beneficial to have the ability to manage the distribution of core configuration and the ability to limit where localizations can be done.

The following diagram illustrates a Microsoft Dynamics CRM Solution being used to push a level of consistency to local areas while also providing the local instance with the ability to support additional, local customizations.



Master data management

Business need

Some organizations are divided by business area (such as retail customer business, professional services business, and partner management). While all business areas may reference a common set of core business information, each area also can overlay that core information with its own area-specific information.

Typically, management in this way can benefit different types of data, each with different needs:

- *Reference data*, which describes the broader world (for example countries, currency codes) or the business (for example branch offices). This information typically changes infrequently, and reflects a broader change.
- *Transactional data*, which describes or results from an action by or with the business. This is important or infrequently changing data t (e.g. customers, authorization levels) that may require change management.

Implementations spanning multiple areas can also arise via the gradual evolution of business systems across which individual system components are replaced to avoid a 'big bang' replacement of an organization's entire system in a single effort. This type of evolution can lead to organizations having multiple integrated systems, possibly including multiple Microsoft Dynamics CRM implementations, but potentially other non-Microsoft Dynamics CRM applications as well.

In these types of scenarios, it is important to maintain the data that is common across the different components and particularly critical is managing changes to that data. While there are different approaches for accomplishing this goal, many scenarios benefit by having a "master" for certain data sets because it provides for change management through that central master data source (master data management, or MDM). Using this approach also ensures that all components have the most up to date information because each references the master.

Solution and constraints

This approach requires that the central master data be synchronized to all instances so that each instance has access to the latest version of the core information. Requested changes to the information can be made directly within the master system (one approach is to delegate a request to the UI of the master system). Alternatively, users can explicitly access the master system or capture the changes in the local instance, with those changes subsequently passed on to the master instance.

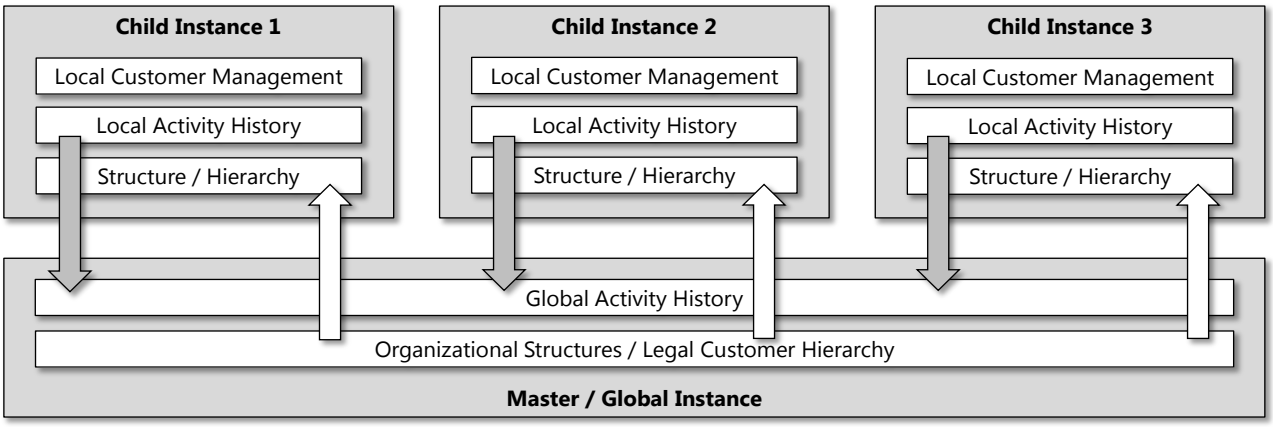
When changes are accepted locally, changes to the master data can get out of sync when multiple, local instances request changes to the same data. Using this approach requires that each local instance be aware of and implement security policies that apply to users changing the master data.

On the other hand, requiring that changes be made centrally can provide for centralized change control. For example, anti-fraud checks can be performed to ensure that changes are only made by a central team and not by local teams

that might otherwise benefit from a change, such as a change in credit limits. This would provide a second level of change authorization and verification that avoids the ability for a single person or a group of people who work closely together to collaborate to effect a fraud. Pushing a request to a different, independent team can provide protection against potential collaboration to perform fraud.

Successfully using the MDM model requires that there be a degree of consistency across data models, which helps to ensure ease of data synchronization between instances. This consistency can be achieved across Microsoft Dynamics CRM instances by importing into each instance a single Microsoft Dynamics CRM Solution that implements a common core data model.

The following diagram shows changes being captured, either centrally or locally, with a central instance being used to manage and propagate those changes to other instances. This can occur either in one direction or it can be bi-directional.



A common MDM model is frequently used for maintaining customer information or reference data, for example a hierarchy of legal entities: Larger companies typically consist of multiple, discrete legal entities (e.g. Microsoft Corporation owns subsidiaries that are separate legal companies in other countries such as the United Kingdom, Germany, and so on.) It can be important to record and track data in a formal way as this may have implications on the legal entity with which the business has legal dealings.

Consider the following examples of reference data:

- Organizational structure
- Locations/sites
- Countries/states/counties
- Partners/employees
- Products/services

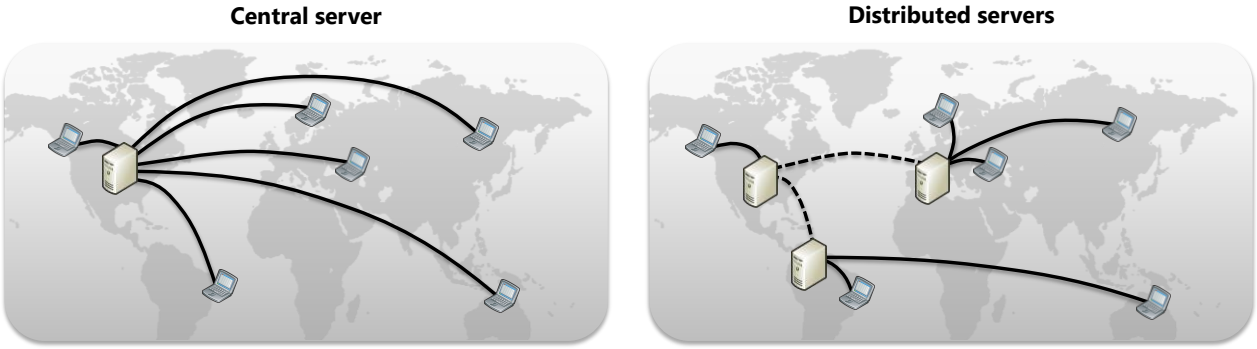
Physical distribution

Business need

When deploying Microsoft Dynamics CRM for an organization, the simplest model to use is a single, centralized instance. However, for business solutions that must support users that are physically distributed over large distances, particularly for global deployments, using a single instance may not be suitable because of the implications (such as WAN latency) associated with the infrastructure over which the users connect, which can significantly impact end-user experience.

Solution and constraints

Distributing instances to provide users with more local access can reduce or overcome WAN-related issues, as the access occurs over shorter network connections. In some regions, using this approach can also reduce the dependency on more heavily used inter-regional connections. Additionally, customers can use local regional instances of Microsoft Dynamics CRM Online to benefit from Internet rather than internal connections. Business scenarios that require global consistency of data, however, also face challenges related to data synchronization.



Security/privacy

Business need

Differences in regional, for example European Union (EU), or national legislation can result in variations in requirements for securing data or maintaining data privacy across the different regions or countries in a deployment. In some cases, legislative/regulatory restrictions make it illegal to host data outside the borders of a country/region, and addressing this challenge is particularly critical in specific business sectors.

For example, consider healthcare sector restrictions on sharing patient information. Some EU regulations require that any health information that is collected about people residing in the EU be maintained and shared only within EU boundaries, while similar data collected about people in the US is kept within US boundaries. Also consider banking sector restrictions on sharing customer information. In Switzerland, for example, regulations make it illegal to share customer information outside of their national boundaries.

In other business environments, it may be necessary to capture different information in each region or country. For example, some countries have legislation forbidding the capture of customers' personal information, such as their hobbies. It is possible to limit users' ability to capture customer data based on what is legal across all legal jurisdictions. In fact, this may be the most appropriate approach to use in a globally connected world in which customers can interact with your organization from around the world.

There are also situations in which using this approach would be prohibitive or overly restrictive on the way business is performed. In these cases, it may be beneficial to allow for variations in information capture to account for differences across multiple regions.

Solution and constraints

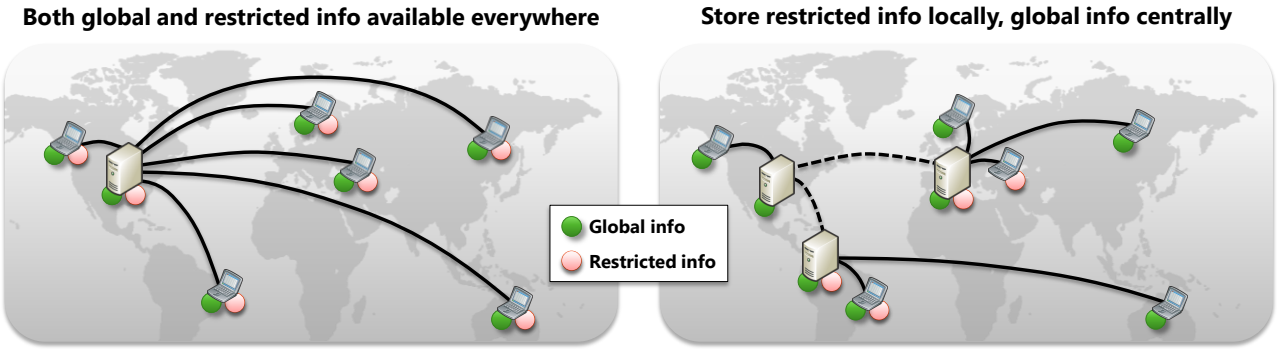
In these types of scenarios, some or all of the data is stored locally, and potentially some of the data is stored centrally. While it may be a valid approach in some cases to simply allow local areas to have their own systems, many larger organizations require that some information be hosted locally, while other elements (such as fault information, knowledge base, organization performance) are centralized or shared.

In these situations, consider deploying Microsoft Dynamics CRM in a hybrid model with either:

- A local, on-premises deployment (for private and secure data) that connects to a centralized deployment of Microsoft Dynamics Online; OR
- A centralized, on-premises deployment that integrates localized instances of Microsoft Dynamics CRM Online within each region, an approach that would simplify the local deployment challenge.

Using hybrid models also can provide for integration with other Microsoft technologies, such as SharePoint, Exchange or Lync. Discussion of integration with different deployment models of other technologies is out of scope of this document, but do note that it is possible.

In either case, user access should be seamless, either by accessing local data on demand or by synchronizing data with a master but offering a unified experience to the end user.



Scalability

Business need

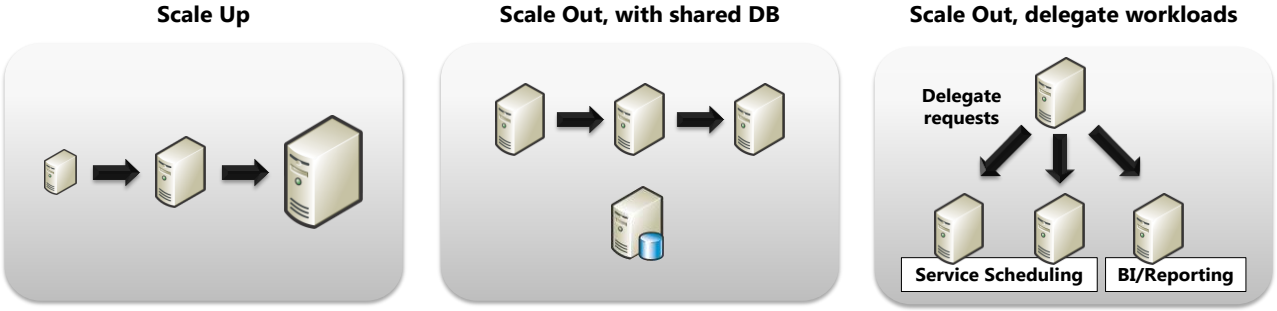
While a single instance of Microsoft Dynamics CRM can scale up and out to support the growth of a customer's business, with very high data volumes or levels of complexity, there are additional considerations. For example, in environments with extreme volumes and/or extensive use of Service Scheduling, scaling up SQL Server can require complicated and expensive infrastructure that is prohibitively expensive or extremely difficult to manage.

Solution and constraints

There are many scenarios in which there is a natural functional split in capability requirements. In such cases, delegating workloads by creating scale out scenarios that are based on these functional splits can provide for higher volumes by using commodity infrastructure. Using this approach has several advantages:

- Provides the same scale with overall reduced cost.
- Aligns with more common enterprise Data Center capabilities in which standard computer specifications may be more in line with the need to scale out rather than to scale up.
- Isolates one area's peaks in demand so that there is no impact on other areas.

Microsoft Dynamics CRM has no native capability to allow this separation of workflow, but for scenarios in which groups of users work independently of each other in operational terms, it may be possible to host the groups on separate Microsoft Dynamics CRM instances and to use reporting to combine results across business areas for management oversight.

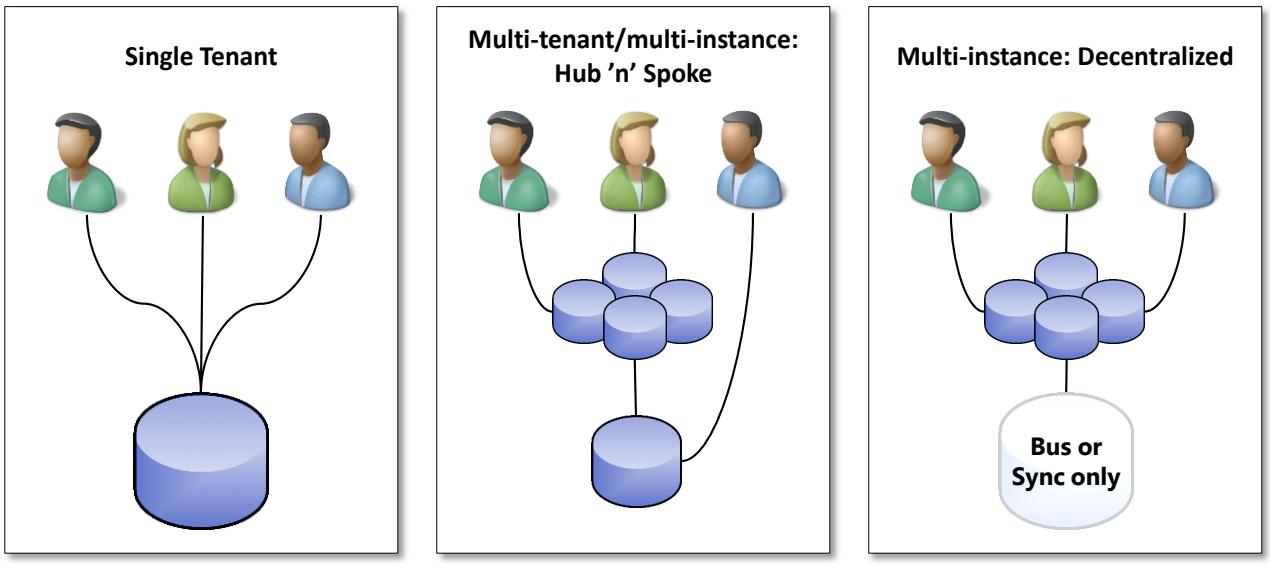


Usage Scenarios: Instance and user access

A review of the different models for using multiple instances of Microsoft Dynamics CRM and multiple tenants, it naturally follows to consider the different ways that instances can be deployed, which would typically be either:

- A single instance containing one or more tenants
- Multiple instances each containing a separate tenant

Configuration can occur in a number of ways with each tenant interacting differently, though some common patterns are shown in the following graphic.



However, remember that there are options beyond the extremes called out above. For example, consider a business scenario that requires functional separation as well as remote deployment to reduce the effects of WAN latency. In this situation, it might make most sense to use a model that simplifies some of the localized deployment effort with multi-tenancy to reduce the overhead of managing large numbers of deployments. One approach here would be to use regional deployments hosting multiple tenants for each country that is local to that region. This approach would bring the regional deployment closer to the end users thereby reducing the potential impact of WAN latency while at the same time allowing each country to maintain its own functional flexibility through the separation provided by multi-tenancy.

While there are countless potential models, the key point here is the need to analyze the details of any specific scenario and then to weigh the benefits and challenges associated with each model to help you identify what may be most appropriate approach.

Multi-tenancy challenges

While there are common solution patterns for supporting multiple instances with existing functionality, common challenges to address include:

- Ensuring a seamless experience in all scenarios and each user touch point
- Developing the custom code that is required to enable some of these patterns
- Ensuring that the functionality provided will continue to take advantage of future product enhancements

CRM Anywhere

In an increasingly mobile and connected world, a key focus of Microsoft Dynamics CRM is to provide capabilities to users wherever and whenever they need it, or enabling 'CRM Anywhere'. When considering a move to this model, there are potential implications for users working in remote or mobile scenarios that need to connect. As a result, be sure to take into account that:

- Many users, for example when using 3rd party mobile applications, will only have mobile access to a single instance.
- While Outlook can connect to multiple Microsoft Dynamics CRM organizations, users can only define one instance for synchronization with the core Outlook data store for Contacts, Appointments, and Tasks.

Despite the fact that users can access multiple instances of Microsoft Dynamics CRM by using the browser client and to an extent through Outlook, it is important to consider the usage patterns when distributing information and functionality across instances and ensuring that the information and capabilities required are available when in mobile situations.

Security

For scenarios in which multiple instances are used in a way that individual users can access more than one instance as part of their role, it is important to allow for the way that their security access is managed. In these scenarios, users who require access to multiple organizations would need their user record configured in each of the instances that they can access.

These users would also need to be assigned security roles in each instance. An approach that often can be useful is to set up asymmetric role access. For example, salespeople may be able to access and edit information about customers in their own region while at the same time for awareness having read only access to customer information in other regions. This can be implemented through provision of different security roles in each instance but would need management of the assignment of roles across instances.

Multi-tenancy provides a sandbox model that can restrict and prevent code from one tenant interacting with data from other tenants. In addition, for an internally focused implementation (in which the target audience for each tenant is part of the same company) that includes trusted code implementations developed in-house, allowing tenants to interact with other tenants' data may benefit efforts to provide a synchronized or coordinated implementation.

Social computing

Using social computing approaches can provide customers with a big advantage in collaborative business scenarios. However, in larger deployments, particularly those that straddle regional or national boundaries, using multi-tenancy has both limitations on and benefits to collaboration via social computing.

One challenge of social computing is the need to maintain an appropriate level of control over information sharing for more secure information types (for example customer information). By using multiple tenants or instances, you can limit information sharing so that it occurs only within natural boundaries, for example by using local country instances

to maintain country specific data. Using this approach can help with security or privacy concerns by preventing the accidental release of sensitive information to a broad audience.

However, this approach can also limit collaboration because using multiple, distinct instances prevents information sharing across instances via social computing features. As a result, for scenarios in which it makes sense to use a multi-instance or multi-tenant model, be sure to take into account the potential impact on collaboration and information sharing using social computing functionality.

Scalability

One key difference between using multi-tenancy and creating multiple instances relates to resource sharing, and the rationale for dividing a solution into multiple organizations will affect the selection of the approach that his most appropriate.

Multi-tenancy is designed to provide the ability to support multiple, lower-demand organizations by hosting multiple tenants on a shared infrastructure. With this approach, each component of the application must be able to accommodate the load generated by all the tenants supported by that instance, which leads to several implications:

- The Microsoft Dynamics CRM web servers must cache the configuration metadata of all of the tenants as well as the details of the users in each tenant, which can:
  - Have memory implications
  - Result in slower performance if the cache is flushed out

  While increasing the memory in the web servers can reduce the associated impact, this also raises the corresponding cost of each web server.
- Jobs that run asynchronously (such as those associated with asynchronous plug-ins and workflows) are queued for processing by the shared Asynchronous servers, which can:
  - Affect the speed at which tasks are completed
  - Have an impact across tenants depending on differences in usage patterns. For example, one tenant may be designed so that a lot of tasks are offloaded to high volume, low priority workflows because the speed at which tasks are completed is not important. Another tenant may be designed to use asynchronous tasks only for a small range of actions with the anticipation that such tasks will occur relatively quickly. In practice, running these two solutions on a single deployment instance means that the workflows of the second solution will not occur as quickly as anticipated because of the volume of tasks from the first solution
- The load on specific database server resources, such as the tempdb, will affect overall performance. Where complex queries are occurring, such as from reporting, there is a greater potential for one query to impact the throughput (for example by locking a table for a query, which blocks other quicker writes until completion of the query) or the performance (for example because of the impact on disk I/O for large volumes of transactions) of other work.

  Note that this can be mitigated by using different SQL Server instances for different organization databases.

At its core, the multi-tenancy architecture is well designed to support hosting several smaller organizations on shared hardware, which makes good use of commodity hardware that otherwise have low usage. Multi-tenancy is also well designed for scaling up as organizations grow in usage volumes.

However, as organizations grow in volume, the advantages of using multi-tenancy rather than separate instances become less pronounced. In fact, at a certain point using separate instances can offer greater advantages (such as allowing a physical distribution of separate tenants) without introducing significant overhead because existing resources in a different configuration are sufficient and are simply distributed and used differently. For example, 4

web servers supporting multi-tenancy volumes could be divided across 2 deployments, providing each with a total of 2 web servers).

A question that often follows relates to whether or not it is possible to physically distribute the web servers more locally to the end users while maintaining the ability to centrally manage the SQL Server instance. While Microsoft Dynamics CRM is heavily optimized to reduce the dependency on a high performance link between the Microsoft Dynamics CRM web servers and the end clients, it does expect a high performance link between the web servers and the SQL Server instance. As a result, it is highly recommended to co-locate the web and database servers rather than split them over a distance, and therefore important to note that use of multiple instances is required in order to locate web servers more locally to the end users.

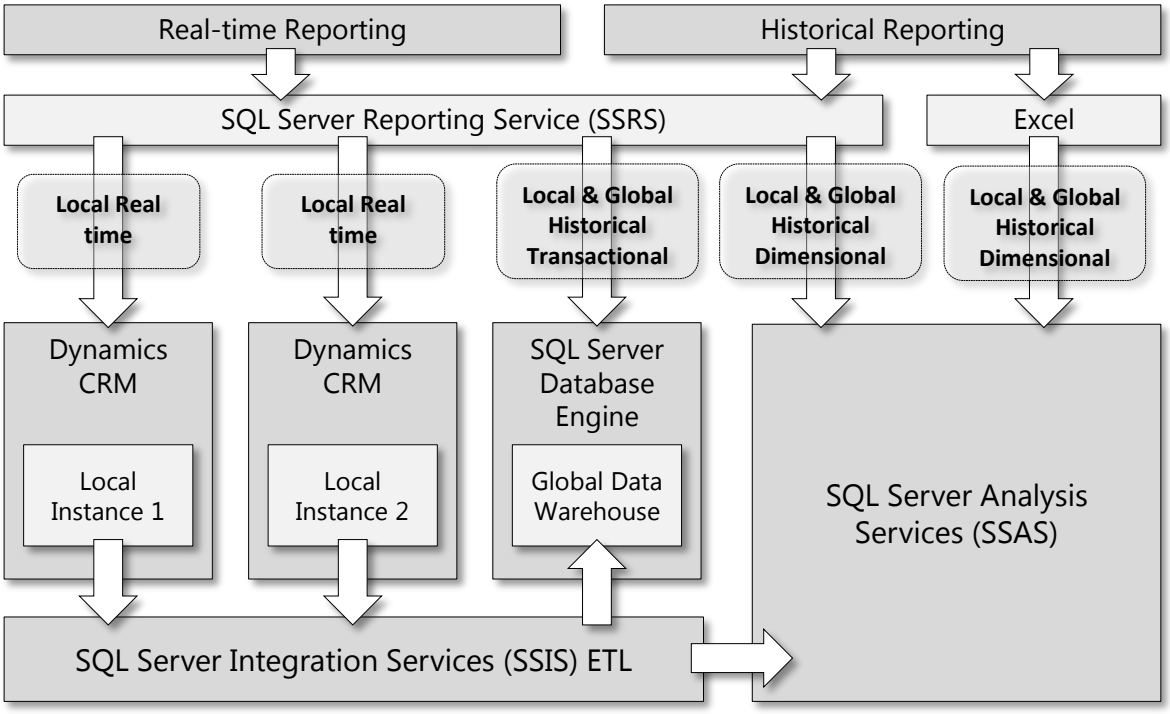
Business intelligence

An often cited reason for maintaining a single Microsoft Dynamics CRM instance relates to the ability to share business intelligence across organizations. Frequently, this is the only requirement that data be aggregated across organizations, while day-to-day operations are limited to data within a user's business area or local region.

In these types of scenarios, it is often possible to enhance the end-user experience by providing a local instance for use in daily operations to each area while other BI capabilities manage the need for cross-instance reporting where there are tools designed explicitly to support exactly these scenarios.

Using a centralized data warehouse can simplify coordination by offering combined reporting to management or access to data in an anonymous format as necessary to avoid privacy or legal concerns. In each case, this can significantly simplify the operational system design while fully meeting and in some cases simplifying the design that would be needed to meet the BI requirements, for example by making the data in the data warehouse anonymous, there may be no need for security controls on the aggregated information.

For on-premises solutions, it may also be possible to perform reporting directly across multiple distributed databases by using techniques such as linked databases, but there are associated performance considerations in scenarios that involve large data sets or slower links between database servers.

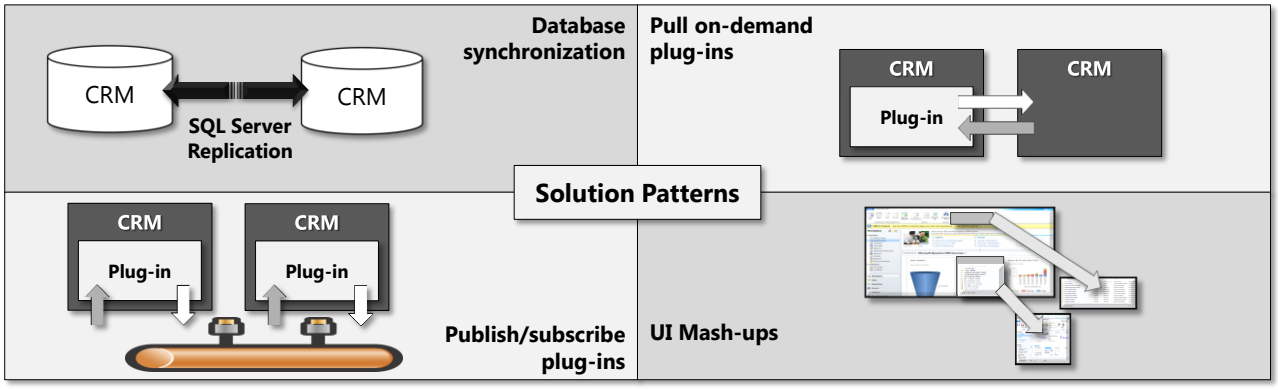


Solution patterns

While taking advantage of multiple instances and tenants can offer benefits, typical enterprise usage also brings with it the challenges associated with managing the separation, unlike managing a single instance with a high-level of consistency across the application. While these challenges can manifest in a number of ways, there are patterns for addressing the associated issues by using features available in Microsoft Dynamics CRM and in the broader Microsoft technology stack, as described in the following table.

Feature	Functionality provided
"Solutions"	Easy management of metadata across instances
Multi-tenancy	Use by independent organizations of the same physical infrastructure and deployments with scale-out across SQL instances
Single sign-on	UI-level integration across instances and systems, creating a more seamless solution avoiding multiple authentications
URL addressable UI	UI-level "mash-ups" to access objects across instances
Plug-in event model	Ability to intercept and alter requests and data
Search	Use of Microsoft search technologies to allow for richer and unified querying and discovery of information across multiple instances
BI	Use of BI tools such as a data warehouse to provide for cross-tenant analysis

The following graphic illustrates various solution patterns that can be leveraged to help manage separate instances or tenants.

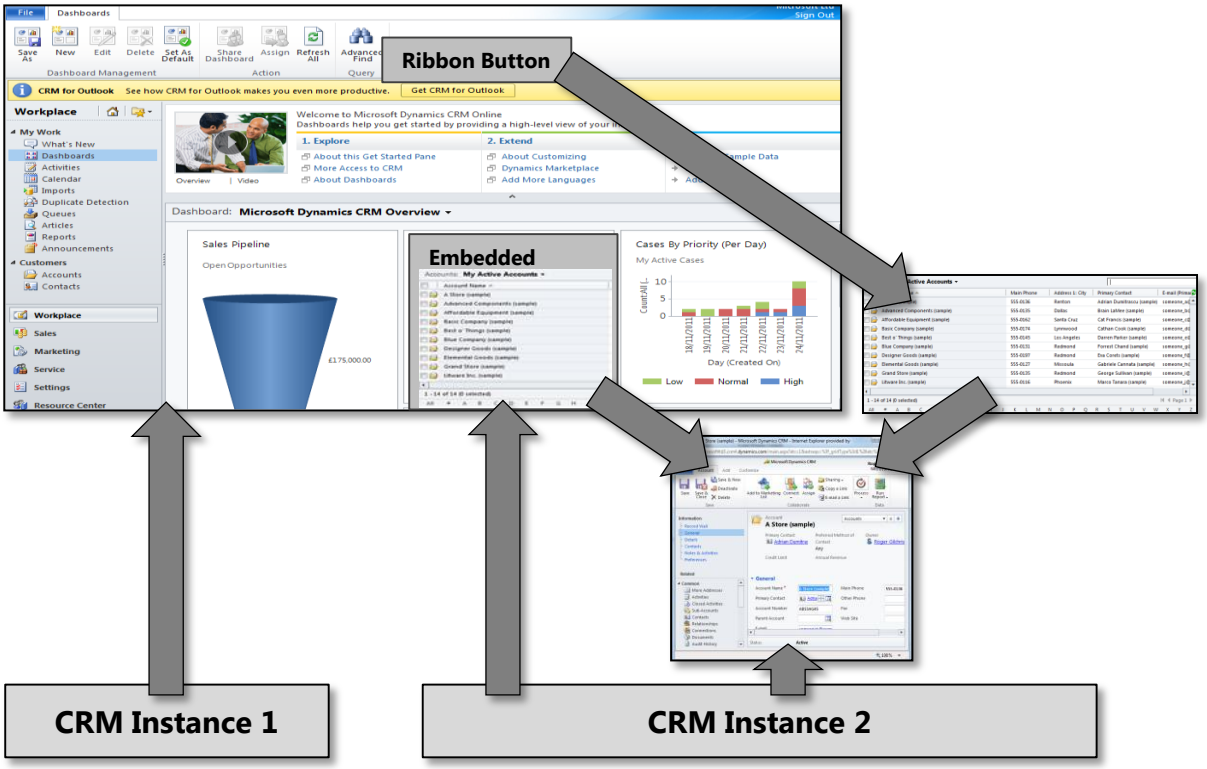


The following sections provide additional detail about each of these solution patterns.

Pattern: UI mash ups

For scenarios in which data is separated across instances, it may be sufficient to provide access to that information or capability by using UI mash ups.

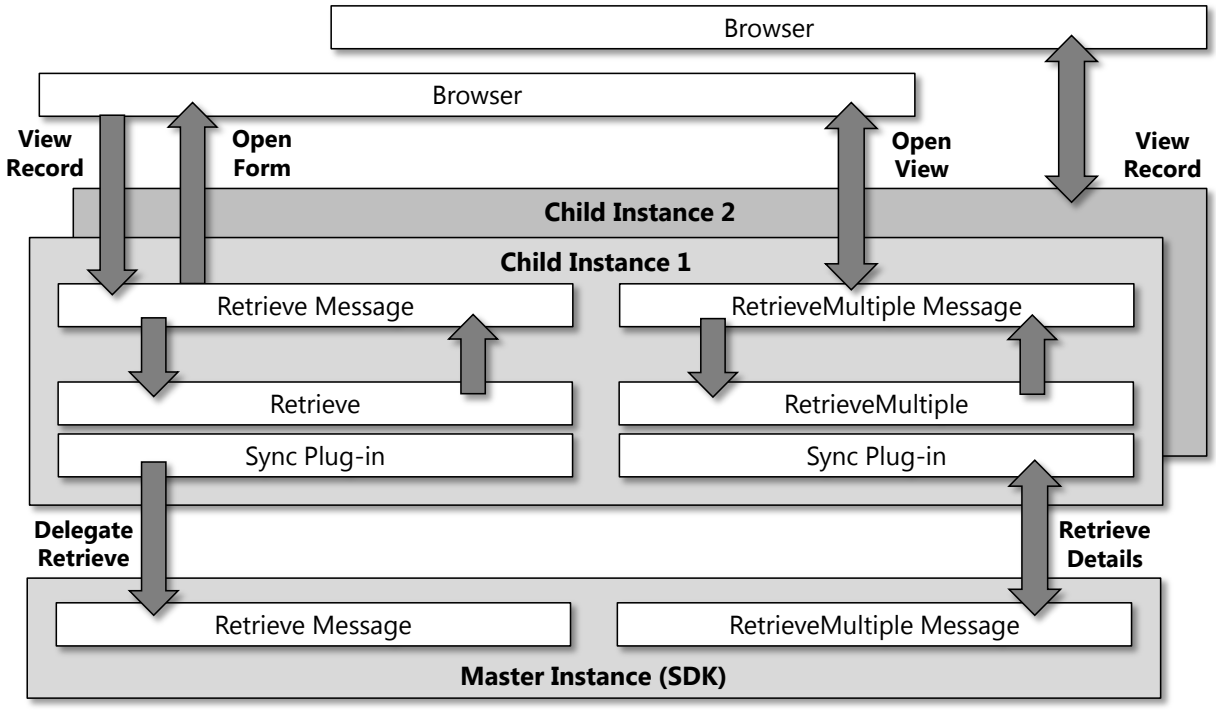
Topic Area	Detail
Usage	▪ Link to UI at other instance using URL addressable pages: ▪ Directly embedded in frames e.g. Dashboards or iFrames, or ▪ Launched as new windows
Requirement	▪ Relies on single sign-on
Caveats	▪ Not always seamless user experience: ○ If different system user records as owner on each side ○ No support for Advanced Find ▪ Can't add relationships between information across instances
Considerations	▪ Security: each user is accessing multiple tenants, so that user will need to be provisioned in each of the tenants with potentially different security roles ▪ Lookups can offer a disjointed experience, especially in situations when: ○ A series of accounts related to a particular user is shown from another tenant users follow through a link on the account form to the owning user, they would be taken to the system user record in the other tenant, which may not be the same user record they started with in the original tenant ○ Users are accessing overlapping data sets ○ Scenarios in which completely different data sets or capabilities are used avoid these types of challenges



Pattern: Pull on-demand plug-ins

Providing a more seamless experience can be achieved by using plug-ins to access information on demand and blending them into the user experience, though this will require additional code to implement.

Topic Area	Detail
Usage	▪ Delegate requests to other instances: ▪ Enables also selective delegation of creates/updates/deletes ▪ Allows for more consistent experience than UI mash ups as supports access via the web services and uses the same UI from the current tenant.
Requirement	▪ Relies on single sign-on
Caveats	▪ Easier for augmenting partial than full record information o Needs a representative instance on requesting side to retrieve a record(otherwise platform step fails as no record exists for record id) o Full Query not easy to delegate (the results for multiple records have implicit actions to open each record, but the link to the record fails as it is not known in the local tenant for example); augmenting result sets with individual fields easier ▪ Doesn't allow for full querying through standard UI o When querying for 'accounts with balances > $100k' if the balance field is only retrieved on demand within the plug-in, the Microsoft Dynamics CRM platform step cannot perform the filter internally to limit the results set by balance o Discovery of new records across multiple orgs therefore need another mechanism to allow full querying e.g. search.
Considerations	▪ Need to consider all the potential interaction points to allow for all the: o Different events that can occur e.g. create/update for record changes, retrieve/retrievemultiple/execute for querying o Ways in which data can be accessed from the system. Web services cannot intercept all possible interactions, and in an on-premises solution the SQL Server filtered views do not trigger plug-ins allowing data manipulation of the results. When the Outlook offline client is used, even with a Microsoft Dynamics CRM Online deployment, the data synchronized to offline can be accessed via SQL Filtered Views from the offline database.



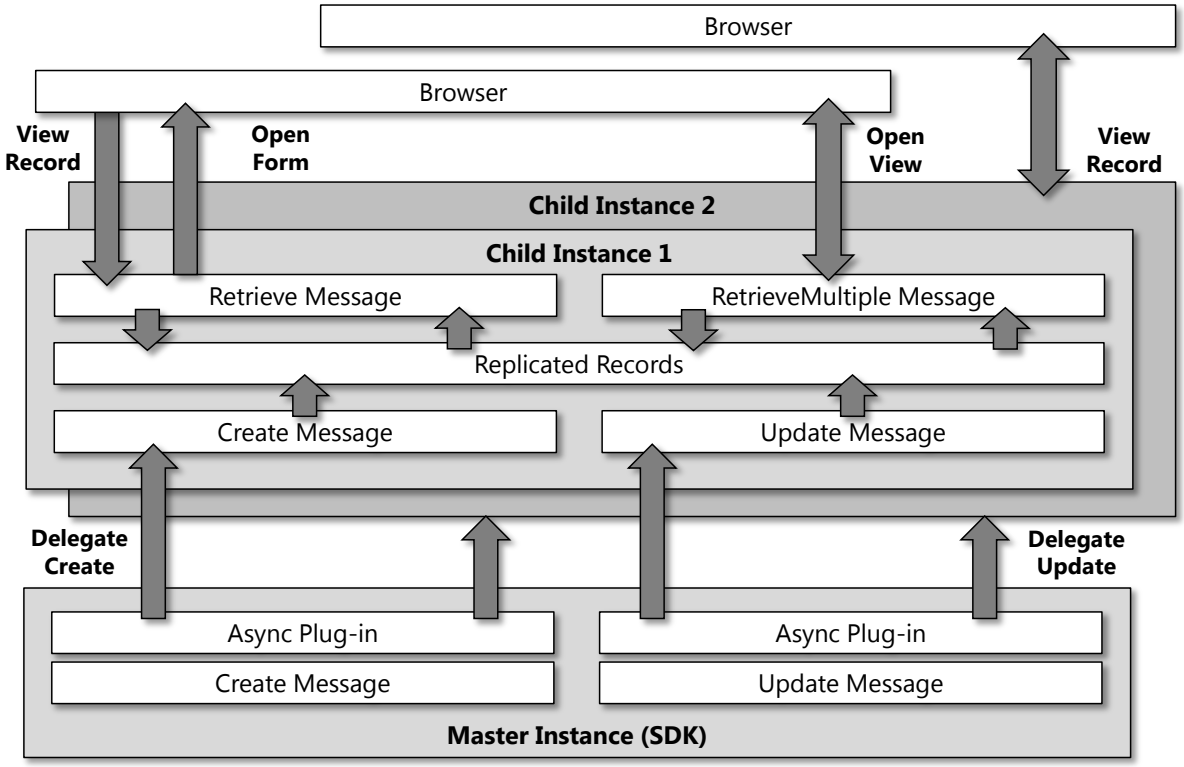
Pattern: Publish/subscribe by plug-ins

Instead of accessing information from other tenants (either multi-tenant or on a separate instance) on demand, synchronization can be performed by using plug-ins to push changes as they are made.

This can be done on a publish/subscribe model, in which the listening instances can register to receive notifications from other instances as changes occur. Using this approach, the types of synchronization that can occur include:

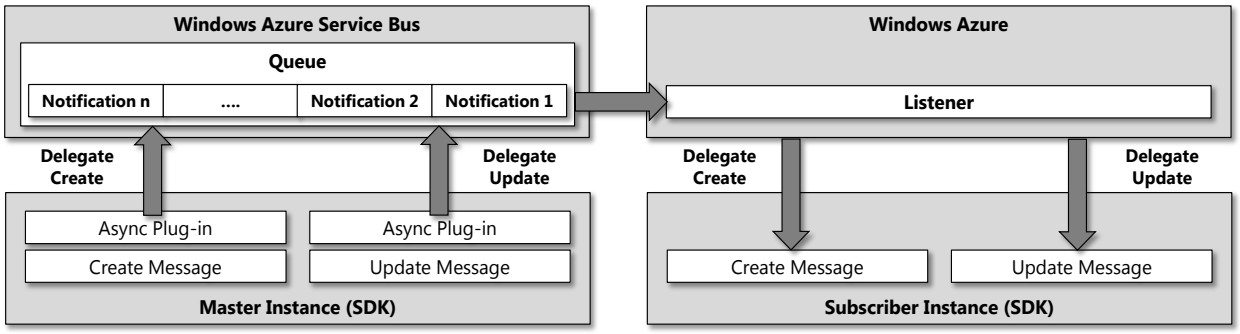
- 2 way synchronization
- 1 way synchronization from master
- Separate masters for different information

Topic Area	Detail
Usage	▪ This pattern can be achieved through: o Replication using plug-in model o Plug-ins to delegate create/ update/delete requests to instances: • Can cater for partial matches i.e. replicate only some of the data • Differing versions o Option to apply rules for subscription
Requirement	▪ Consistent management of implementation between solutions
Caveat	▪ Need confirmed write, to avoid data loss; need to judge the complexity over accepting loss that can be recovered by later write
Consideration	▪ Can be provided through ESB; initially more complex than direct model for smaller implementations, but provides greater decoupling and better suited to richer models



Pattern: Publish/subscribe via service bus

An extension of the pattern above would be to use integration with the Azure Service Bus. This approach can be used to push events directly to the service bus as they occur on the source instances without code on the source system. To register interest and receive the notification events, a listener would need to be implemented to receive the events and pass them on to the subscriber instances. This listener would need to be hosted outside of Microsoft Dynamics; Windows Azure provides a perfect home for this in a cloud based model.



Pattern: Synchronization

Another approach is to perform synchronization between tenants using an external tool. With this model, the external tool monitors Microsoft Dynamics CRM tenants for changes and then pushes the changes to the other tenants as appropriate. This pattern can also be used to synchronize with systems based on technologies other than Microsoft Dynamics CRM, but the appropriate schema and data transformations would be required.

There are a number of common tools used for this purpose including:

- SQL Server Integration Services
- BizTalk Server
- Dynamics CRM Connector

- Third Party Tools

Many of these tools provide for deploying either on-premises or cloud solutions. While SQL Server filtered views can be used for read only detection of changes, this is only supported for on-premises deployments. As a result, it is recommended that the Microsoft Dynamics CRM web services are used both for querying from and for pushing of changes to Microsoft Dynamics CRM, as taking this approach provides the most deployment flexibility.

SQL Server Integration Services

SQL Server Integration Services (SSIS) has the advantage that it is part of SQL Server, and therefore can be configured as part of the existing SQL Server installation. Even though SSIS is primarily focused at moving data between databases, it provides the ability to integrate with external web services. There are 3rd party add-ons for SSIS that simplify the connection to Dynamics CRM Web Services. It is possible therefore to use SSIS to provide synchronization services between Dynamics CRM instances.

BizTalk Server

In a similar way, BizTalk Server enables synchronization processes to be configured. BizTalk is able to integrate with WCF Web Services, Dynamics CRM provides schemas to assist with connecting to the Dynamics CRM SDK Web Services.

Dynamics CRM Connector

There is a tool called the Connector for Dynamics Connector and an additional adapter that allows for the synchronization of data between two Microsoft Dynamics CRM 2013 organizations. The adapter allows for communication between two endpoints that exist on any authentication/hosting environment (on-premise, online, IFD, etc.)

This means that you can leverage this new adapter to move Microsoft Dynamics CRM 2013 data between test and production servers, different instances or from on-premises to the cloud. The Microsoft Dynamics CRM 2013 Instance Adapter is freely available here from the Microsoft Download Center and is offered as an unsupported add-on to Connector for Microsoft Dynamics.

Third party tools

There are also a range of third party tools that offer similar services. These range from tools focused on Dynamics CRM to more broadly focused ETL/Synchronization tools, many of which have Dynamics CRM specific adapters to assist with setting up and connecting to Dynamics CRM.

Deployment models

As has been noted above, many of the patterns above can be used in either on-premises or online deployment models. As a result, it is also possible to implement hybrid models, which entail a combination of online and on-premises instances. Hybrid models can be particularly effective in physically distributed scenarios, using Microsoft Dynamics CRM Online to provide capabilities to remote deployments, delegating the responsibility for the underlying deployment to Microsoft rather than needing to manage the installation of a local instance in remote regions of the business.

Another increasingly popular approach takes advantage of private cloud technologies such as those provided by Microsoft System Center 2012, with which organizations can deploy Microsoft Dynamics CRM by using the underlying virtualization to provide both isolation between instances and flexible resource management to account for changing demands. Particularly when managing higher volume Microsoft Dynamics CRM solutions, a hybrid model can provide a great degree of flexibility for future deployments.

Summary

We provide the capability to model various types of business processes in Dynamics CRM – including the ability to extend the deployment to multiple CRM tenants or instances as needed based on business requirements.

Common challenges in large scale deployments that using multiple Microsoft Dynamics CRM organizations can assist with include:

- Functional localization
- Master data management
- Physical distribution
- Security and privacy
- Scalability

Often, the best way to address these types of challenges is by using multiple Microsoft Dynamics CRM tenants and determining which if any tenants can share an infrastructure and which might require a dedicated infrastructure. For solutions that do incorporate separate Microsoft Dynamics CRM instances (or tenants), a second major consideration is whether or not there are requirements for integration or interaction between the different tenants and instances.

In conclusion, be sure to remember that using multi-tenant and multi-instance deployments can help to solve large scale enterprise challenges, but the overall benefit requires co-ordination and collaboration between the instances. Though Microsoft Dynamics CRM doesn't provide native support for this co-ordination, there are customization patterns that can help to address these challenges.

Appendix A: Additional resources

For more information about using multi-tenancy in Microsoft Dynamics CRM 2013 to address challenges in enterprise business environments, see the following technical resources.

Related implementation topics for IT Professionals

Implementation and Installation

- Microsoft Dynamics CRM 2013 Implementation Guide
  - Download
  - View online

Microsoft Dynamics CRM "Solutions"

- ALM for Dynamics CRM 2011: CRM Solution Lifecycle Management

Pattern: UI mash ups

- Sharing Data across Microsoft Dynamics CRM Deployments (Microsoft Dynamics CRM 4.0)

Windows Azure

- Introduction to Windows Azure Integration with Microsoft Dynamics CRM

Related implementation topics for Developers

Microsoft Dynamics CRM "Solutions"

- Package and Distribute Extensions with Microsoft Dynamics CRM Solutions

Pattern: UI mash ups

- Open Forms, Views, and Dialogs with a URL
- Analyze Data with Dashboards
- Customize the Ribbon for Microsoft Dynamics CRM

Plug-ins

- Extend Microsoft Dynamics CRM with Plug-Ins

Azure Service Bus

- Azure Extensions for Microsoft Dynamics CRM