

On Premise Design Plan - 2019

Server Application Update and Versioning - DSR

- There should be notifications when the update is available
- Online
 - o BatchWorker checks for the new version in the shared location and in the network
 - o BatchWorker downloads a package and stores it in the folder for the updates
 - § File storage should be used for storing updates because it is shared between the components
 - o User should confirm the update
 - o Marks the DB that the Update is available and confirmed (BatchWorker)
 - § the user should confirm the update start
 - o Each component will shut itself down and reinstall itself (including the CM)
 - o Once the update process is done the health status should be updated
 - o Alternatively, the admin console should trigger the update.
- Offline
 - o The update is downloaded manually
 - o The user puts the installer into the downloads folder
 - o The Admin Console accepts the update package and stores it
 - o ... the same as the online version
- Rollback
 - o There should be a history of updates and ability to go to any of it
 - o For now - manual process
- Downloads and puts into the permanent storage
- Indicate that the update is available.
- Each component will update itself on its own (maybe)
- Health dashboard is required
- Shared Updates location (centralized)
 - o Each component will have its own access to the centralized storage
- Getting older versions in the new installation - is it possible?
 - o New users should get the latest version for the 1st installation
 - o After the 1st installation we should store the installation package in the storage.
 - o Will require the support managers training
- Currently the Admin Console worker is supposed to be the updater but it conflicts with the admin console role. Can we move the Call Home and update functionality to Admin Console
 - o The Batch Worker downloads the files and marks the DB as ready for update
 - § The BW will delete all intermediate updates so the user will not be able to install them. User will be able to install the newest version only.
 - o Admin Console will control the installation

- Will the updates be located in the central storage or on each machine separately (it will have to download them somehow)?
- o All updates should be in the file storage which is accessible for all components
- o We don't have to encrypt the updates files
- o The key is stored in each component configuration string (please see above)

Database Update and Versioning

- Nice to have option: install the SQL express for the user and start it
- Change the Utopia to use migrations and code first at some point
- o We will use code first and migrations.

----- Old Notes -----

Researches to be done:

- Research for feasibility: - done
- o drag out event from the browser control (including Outlook)
- o Important to have
- Research for options (BE):
- o Install Shield installer and updater for Desktop when the pipe is small
- Research for solution of changing IP address (BE):
- o Self-hosted ClickOnce installer and updater that checks for updates manually and provides a link to download
- Research if Mac updates the applications of the same name and different versions, especially, for non-store apps
- Research: can we implement Drag and Drop for the Windows taskbar icons (it works in Mac) - lowest priority
- Research: running indexer (Elasticsearch) on a laptop and its memory consumption
- o Possibly it can be excluded for small accounts
- SQL Server Express: can we include it into the installation package?
- Multi-instance data migration for Desktop: migration and integration.
- o Only for Casselle, possibly they will go to Cloud
- Research the encryption so that we can use the main key and recovery keys
- Proof of concept: self-updating service that starts the installer and then itself
- Research: if dotless can be used in the desktop version

Client UI

- Will be available for Mac (phase 2) and Windows (phase 1)
- Keep the existing functionality to support d&d, printing, scanning, watch folder, office addins, managing check-in/out
- Change the UI to Web completely (except settings and Addins UI)
- o There is no big value in changing the Settings and Addins UI to Web
- o The Store Window can be a value for attracting clients to the Utopia and change the name of scanned files
- § Should be enabled by a user setting
- o Store dialog design is required
- Discussed Items:

- o Switching modes: A button on the top of the form that hides everything and shows the browser window
- o Implement dragging files on the SideKick taskbar icon so that we can avoid showing the Orb - (not many windows users use it) ?
- o Remember the last selected mode: Maximized and Minimized
- o Authentication:
 - § Reuse the authentication screen from the SideKick
 - § Do not use the Web Auth screen
 - § Change the browser to refresh the authentication using the refresh token
 - § Track the url or JS event for logging out - display the sign in screen from the SideKick
 - § The SideKick sign in UI has a server name/IP address selector
 - § There should be a mode selector for full-sized mode and limited mode. Prompt can be here as well.
- o Disconnected mode for the web and limited view
- o The front end side will be done by the EFC team
- o For the future: Move the settings, addons etc. to the web part
- o Configuration
 - § Should include the OCR running schedule (currently it does not support the schedules) - admin console
 - §

Server Side

- Servers discoverability of the desktop can be postponed for phase 2
- Integration implementation (store dialog and population) - phase 2
- Requirements
 - o The Desktop 2.0 should be able to handle up to 1000 users
 - § The requirement is fulfilled
 - o We don't need to care about the Load Balancing (it will be done by customers)
 - o We need to make sure the system components will be configurable and will be able to handle load balancers configuration
 - o Installation must consider that different components can be installed on different machines
 - § Current Desktop configuration is 2-3 different machines: OCR, Server Services + Indexing service
 - o Instructions documentation should be provided
 - § Should include the list of load balancers that will work
- Scaled components:
 - o Database
 - o Application server
 - o Batch Worker can be included into the installer
 - o Index server (ElasticSearch)
 - o OCR server
 - o Image server
 - o Admin Console (native or web-hosted)
- Non-functional requirements:
 - o Indexer limitation for memory consumption (to be defined)
 - o OCR server limitation for threads number, CPU usage
 - o Must be able to work offline completely
- Discussion

- o Switching to self-hosted Web API
- o OCR should be the separate installer because many of the customers will not have it
- o Must work offline (including the licensing)
- o There should be a provider that works on the local file system
- § The files should be encrypted individually
- § It should not take long
- § There should be a way to avoid windows file system number of files limitation
- § There will be no ability to map different locations for file storage
- o Backdoor: need a research encryption best practices to have several keys that can unlock.

Licensing:

- Batch worker will need to call home periodically for the account updates (contains features and keys that will expire)
- Batch Worker will be a kind of the licensing server
- Batch Worker should be able to update itself
- The user should be able to update the application package by downloading it and applying.
- Application package can be downloaded independently from the update package
- The Account package will be provided annually
- Obfuscation is required

Installation and configuration

Requirements

- Installer should have a link to SQL Express.
- Updates should be optional. The update package should be downloaded for the account and should be downloadable separately.
- Will need an installation wizard
- We will need a configuration wizard that can be reused to set/edit configuration
- Dashboard (admin console) should be combined with the installation wizard
- Admin console should be a prerequisite for the installer in order to avoid multiple Admin Consoles installations.
- Will need to have several installation options

o Typical

- § Should be a single machine, should work on laptops and should be as simple as possible
- § Will still include the Admin console full functionality

o Custom

- We need an MSI for each individual component and all inclusive MSI that includes all individual components MSIs
- User is able to install the components without signing in and licensing
- o Obfuscation is nice to have, can be phase 2
- Will need to provide ability to update the server when it is offline
- o Using the installer
- o Should prevent a component from running if it is not up to date
- § Can be done by a component
- § Will be done through a heartbeat table for each component
- § The version should be tracked by the admin console
- Backup - ask JR and Jesse about it
- o Should be done before each update

- o Should have its own folder to store data
- o Should not include the files

Admin Console

· Requirements

- o Should include the admin email setup for notifications
- o If the BatchWorker is not able to phone home then the user should be able to provide an account data package and the account data hash
- o The user will be able to download the account data package from the EFC server
- o Remote configuration
 - § Provide the configuration string for a remote machine. Remote machine config wizard will be able to accept the string and configure the services by it.
 - § Can be used in the installer as well
 - § Updating configuration is a challenge: the user needs to know that each component configuration should be updated
 - § If the component starts and there is no configuration it should start a configuration manager (if possible for a component)
- o Includes the UI to check health and configure
- o Each Utopia configuration item should be configurable except the SecureDrawer
 - § Please see the UtopiaConfiguration class for details
- o We can allow them to enter their own encryption string
 - § In this case, there will be no way to restore if the key is lost
 - § Generate a key for them
 - § generate recovery keys for them (like 2FA in GitHub)
 - § We need to have 2 options for encryptions keys: auto generated and provided manually
- o It should be a separate component that can be installed on a different machine
- o Should provide access to the older installers in case of the version downgrade
 - § Sometimes customers downgrade but more often they just need an older version of a particular component
- o If the service is not able to phone home it looks at the installation packages and sends an email to the admin about the account data packages expiration
- o Should show all the components and their versions
- o Running status of the components
 - § Will work through the DB
- o Will need to add some checks for cheating

Client application update

- We might switch the application to use the same installer (Install Shield or ClickOnce or anything else). It depends on the installers research.
- Current (ClickOnce)
 - o Locally hosted ClickOnce application (on the server which will get the updated ClickOnce installer once it is updated by itself)
 - o The challenge is to switch the client from Cloud to local server

Server Application Update and Versioning

- There should be notifications when the update is available
- Online

- o BatchWorker checks for the new version in the shared location and in the network
- o BatchWorker downloads a package and stores it in the folder for the updates
- § File storage should be used for storing updates because it is shared between the components
- o User should confirm the update
- o Marks the DB that the Update is available and confirmed (BatchWorker)
- § the user should confirm the update start
- o Each component will shut itself down and reinstall itself (including the CM)
- o Once the update process is done the health status should be updated
- o Alternatively, the admin console should trigger the update.
- Offline
- o The update is downloaded manually
- o The user puts the installer into the downloads folder
- o The Admin Console accepts the update package and stores it
- o ... the same as the online version
- Rollback
- o There should be a history of updates and ability to go to any of it
- o For now - manual process
- Downloads and puts into the permanent storage
- Indicate that the update is available.
- Each component will update itself on its own (maybe)
- Health dashboard is required
- Shared Updates location (centralized)
- o Each component will have its own access to the centralized storage
- Getting older versions in the new installation - is it possible?
- o New users should get the latest version for the 1st installation
- o After the 1st installation we should store the installation package in the storage.
- o Will require the support managers training
- Currently the Admin Console worker is supposed to be the updater but it conflicts with the admin console role. Can we move the Call Home and update functionality to Admin Console
- o The Batch Worker downloads the files and marks the DB as ready for update
- § The BW will delete all intermediate updates so the user will not be able to install them. User will be able to install the newest version only.
- o Admin Console will control the installation
- Will the updates be located in the central storage or on each machine separately (it will have to download them somehow)?
- o All updates should be in the file storage which is accessible for all components
- o We don't have to encrypt the updates files
- o The key is stored in each component configuration string

Database Update and Versioning

- Nice to have option: install the SQL express for the user and start it
 - Change the Utopia to use migrations and code first at some point?
 - o We will use code first and migrations. DSR will work on WBS and estimate
 - o The available team can start working on it (possibly, DSR, after the Pandora and Pangea)
 - o We will need a training after it is done (if DSR does it)
-

Revision #1

Created 18 July 2022 23:36:03 by Bryce Holloway

Updated 9 February 2023 20:20:50 by Bryce Holloway