Trusted Content Storage (TCS Stack): Decentralized Twitter[1] (Dewitter) App Scenario

Version 0.32 Review Draft - March 18, 2021

Michael Herman
Self-Sovereign Blockchain
Futurist, Architect, and Developer

Trusted Digital Web
Hyperonomy Digital Identity Lab
Parallelspace Corporation

Alberta, Canada

mwherman@parallelspace.net

The publication of this document coincides with recent discussions (January-March 2021) about secure data storage solutions in the Decentralized Identity Foundation (DIF) Secure Data Storage working group (sds-wg) that were taking place during the development of the Confidential Storage specification. This is not a DIF publication, unofficial, official, or otherwise.

[1] Twitter is trademark of Twitter but Twittad also owns it too.
Reference: https://mashable.com/2011/10/10/twitter-tweet-trademark/

TABLE OF CONTENTS

TABLE OF FIGURES

CHANGE HISTORY

Version 0.32

1. Minor edits

Version 0.31

1. Minor edits

Version 0.28

1. Item 23. Personal Agent to Local EDV Server Instance Communications. Added.

Version 0.27

1. APPENDIX C - Licensing. Change from *Creative Commons Attribution 4.0 International Public License* to *Creative Commons Attribution-ShareAlike 4.0 International Public License*.

Version 0.26

1. Item 48. Like a Tweet. Fixed typo.

Version 0.25

1. Item 14. Tweet Like Counter and Tweet Like Account List. Added.
2. Item 48. Like a Tweet. Updated.

Version 0.24

1. Items 3.5-3.7. Additional clarification added with respect to an account (Identity) having a Neighborhood of Followers and a Following.
2. Added Item 15. Stream Key. A Stream Key is a Tweet Key (for this version of the app scenario). Made appropriate changes where needed elsewhere in the document.
3. Item 29. Query a Personal Local EDV Service Instance for a Stream Item. Updated to reference the Generic Replication Pipeline pattern.
4. Item 30. Query Another Personal Agent or List of Personal Agents for a Stream Item. Updated to reference the Generic Replication Pipeline pattern.

ABSTRACT

The purpose of this document is to describe a Decentralized Twitter (Dewitter) App Scenario based on the Trusted Content Storage Architecture (TCS Stack).

The intended audience for this whitepaper is a broad range of professionals interested in furthering the application and use of a highly secure, modular, encrypted data storage solution for use in software apps, agents, and services. This includes software architects, application developers, and user experience (UX) specialists; as well as people involved in a broad range of standards efforts related to decentralized identity, verified credentials, and secure storage.

The work documented here was performed under the auspices of the Trusted Digital Web project in the Hyperonomy Digital Identity Lab of Parallelspace Corporation.

This is not a DIF publication, unofficial, official, or otherwise.

CONTEXT

> *"Sometimes called "reasoning from first principles," the idea is to break down complicated problems into basic elements and then reassemble them from the ground up. It's one of the best ways to learn to think for yourself, unlock your creative potential, and move from linear to non-linear results."*
>
> [First Principles: The Building Blocks of True Knowledge (https://fs.blog/2018/04/first-principles/)]
>
> *"I think it is most important to reason from first principles rather than by analogy. One of the ways we conduct our lives is we reason by analogy. We do this because something was like something else that was done or it was like what other people were doing. It's mentally easier to reason by analogy rather than from first principles."*
>
> [First Principles Method Explained by Elon Musk (https://www.youtube.com/watch?v=NV3sBlRgzTI)]

Purpose

The purpose of this document is to describe a Decentralized Twitter (Dewitter) App Scenario based on the Trusted Content Storage Architecture (TCS Stack). Dewitter is intended to be a completely standalone, decentralized implementation of a Twitter-like platform. Dewitter is not intended to have a physical connection or otherwise with the actual Twitter platform or app.

Intended Audience

The intended audience for this whitepaper is a broad range of professionals interested in furthering the application and use of a highly secure, modular, encrypted data storage solution for use in software apps, agents, and services. This includes software architects, application developers, and user experience (UX) specialists; as well as people involved in a broad range of standards efforts related to decentralized identity, verified credentials, and secure storage.

Organization

This whitepaper contains the following sections:

- Context (this section)
- Problem Statement
- Solution Concept
- Conclusion
- APPENDIX A: Resources
- APPENDIX B: Generic Replication Pipeline
- APPENDIX C: Licenses

PROBLEM STATEMENT

Overview

This section addresses the purpose of the whitepaper through a series of successive evolutions of the Decentralized Twitter (Dewitter) App Scenario based on the Trusted Content Storage Architecture (TCS Stack)[2].

Motivation

The original impetus for this whitepaper includes:

- The Confidential Storage specification being developed by the Secure Data Storage working group (sds-wg) within the Decentralized Identity Foundation (DIF)
- Several sds-wg conference calls in January and February 2021 and related side conversations
- Development of the Trusted Digital Web: Trusted Content Storage Architecture (TCS Stack): Architecture Reference Models (TCS-ARMs) whitepaper.
- Desire to develop a Secure Resource Sharing solution concept that leverages the TCS Stack.
- Desire to develop and prioritize a set of features for version 1 of the Confidential Storage specification.

[2] Michael Herman, *Trusted Digital Web: Trusted Content Storage Architecture (TCS Stack): Architecture Reference Models (TCS-ARMs),* version 0.96, http://hyperonomy.com., March 2021.

Key Concepts

TCS Stack

The TCS Stack is illustrated below.

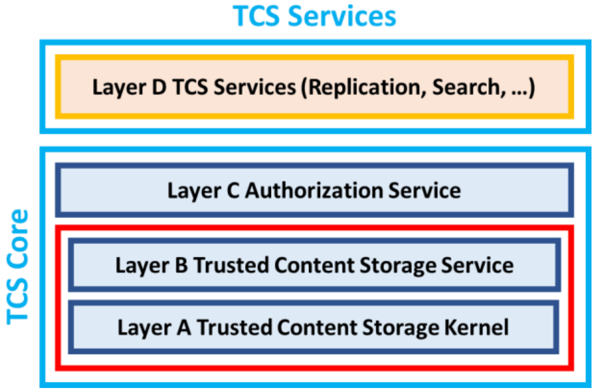

TCS Services
Layer D TCS Services (Replication, Search, …)
TCS Core
Layer C Authorization Service
Layer B Trusted Content Storage Service
Layer A Trusted Content Storage Kernel

Figure 1. TCS Stack

Detailed TCS Stack

The Detailed TCS Stack Diagram expands Layer A Trusted Content Storage Kernel and Layer B Trusted Content Storage Service layers to expose more of each layer's subcomponents (sub-elements).

In Layer A Trusted Content Storage Kernel, the primary subcomponent is the EDV Microkernel.

In Layer B Trusted Content Storage Service, the primary subcomponents are:

a. the protocol handlers (endpoints), and
b. the underlying services needed to integrate them with the EDV Microkernel.

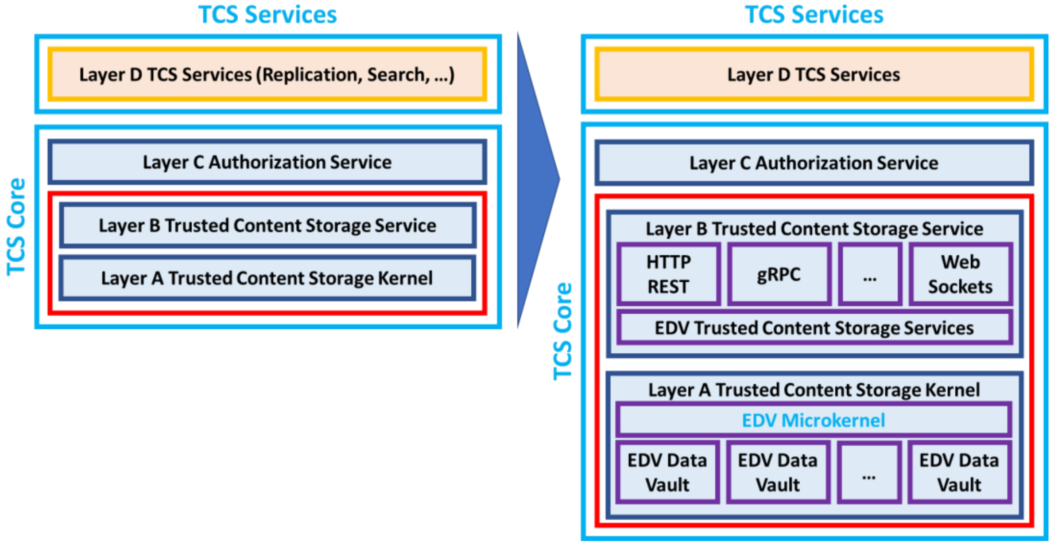

TCS Services
Layer D TCS Services (Replication, Search, …)
TCS Core
Layer C Authorization Service
Layer B Trusted Content Storage Service
Layer A Trusted Content Storage Kernel
TCS Services
Layer D TCS Services
TCS Core
Layer C Authorization Service
Layer B Trusted Content Storage Service
HTTP REST
gRPC
…
Web Sockets
EDV Trusted Content Storage Services
Layer A Trusted Content Storage Kernel
EDV Microkernel
EDV Data Vault
EDV Data Vault
…
EDV Data Vault

Figure 2. (a) TCS Stack mapped to (b) Detailed TCS Stack

Dewitter Decentralized App Scenario

Assumptions, Principles, Requirements, and Other Considerations

The following assumptions, principles, requirements, and other considerations underpin the development of the initial *fully decentralized* Dewitter App Scenario.

General Assumptions

1. General Assumptions
    a. For the baseline description of the fully decentralized Dewitter App Scenario, each actor is assumed to have a local device (i.e. smartphone, laptop, and/or tablet).
    b. There are no servers in the initial variation of the app scenario: Dewitter is *fully decentralized*.

    NOTE: In a subsequent variation of the baseline fully decentralized app scenario, the addition of centralized servers and server-hosted EDV Server Instances will be considered. This variation is called "Distributed Twitter (Distwitter)" because it runs across multiple centralized servers in addition to personal Local EDV Server Instances; that is, Distwitter is not fully decentralized.

Actors, Roles (Personas), and Roles (Followers, Following, Neighborhoods)

2. Actors

There are 3 actors in the initial version of the Dewitter App Scenario.

Actor	**Description**
2.1. Alice	Alice is a Dewitter user with 2 Dewitter accounts (identities): @alice and @alicecooper. Alice uses Dewitter on 2 devices: Alice's Phone and Alice's Laptop. On Alice's Phone, Alice uses both identities. On Alice's Laptop; Alice only uses the @alice identity. When Alice isn't using Alice's Laptop, Alice closes Alice's Laptop.
2.2. Bob	Bob is a Dewitter user with 1 Dewitter account (identity): @bob. Bob uses Dewitter on 1 device: Bob's Phone. On Bob's Phone, Bob uses the @bob identity.
2.3. Carol	Carol is a Dewitter user with 1 Dewitter account (identity): @carol. Carol uses Dewitter on 1 device: Carol's Tablet. On Bob's Phone, Bob uses the @bob identity.

3. Roles (Friends and Followers)

There are 4 roles (friends and followers) in the initial version of the Dewitter App Scenario[3].

Role	Description
3.1. Friend (Friends)	A Friend is a reciprocate social link between 2 accounts (identities), which indicated that the 2 accounts (identities) are acquainted with each other and willing to help each other.
3.2. Neighbor (Neighborhood)	A Neighbor refers to the relationship between accounts (identities) with common interests; for example, accounts (identities) sharing the same Following are a Neighborhood. An account (identity) and all of its Followers and Following make up a Neighborhood.
3.3 Follower (Followers)	A Follower of Alice is someone who has an interest in Alice's activities and wants to receive notifications about Alice's activities.
3.4 Following (People I'm Following)	One of Alice's Following is someone whose activities interest Alice and that Alice would like to receive notifications of their activities.

3.5 Each account (identity) (e.g. @alice, @alicecooper, @bob, @carol) has the following collections associated with it:

a. Friends
b. Neighbors
c. Followers
d. A Following

3.6 Each account (identity) is only interested in (needs to exchange information with) its Neighborhood (i.e. its Followers and Following).

3.7 Each account (identity) is primarily interested in learning about new Neighborhood Tweets since the last Tweet it viewed from each Following account (identity).

4. Roles (Personas)

There are 7 roles (personas) in the initial version of the Dewitter App Scenario.

Role	Description
4.1. Consumer	Consumer has an account (identity). Consumer only reads tweets. Never creates new tweets. Never searches for tweets. Never replies nor retweets to other tweets. Consumer comes in 3 variations based on the frequency of use: Daily Consumer, Weekly Consumer, and Monthly Consumer.

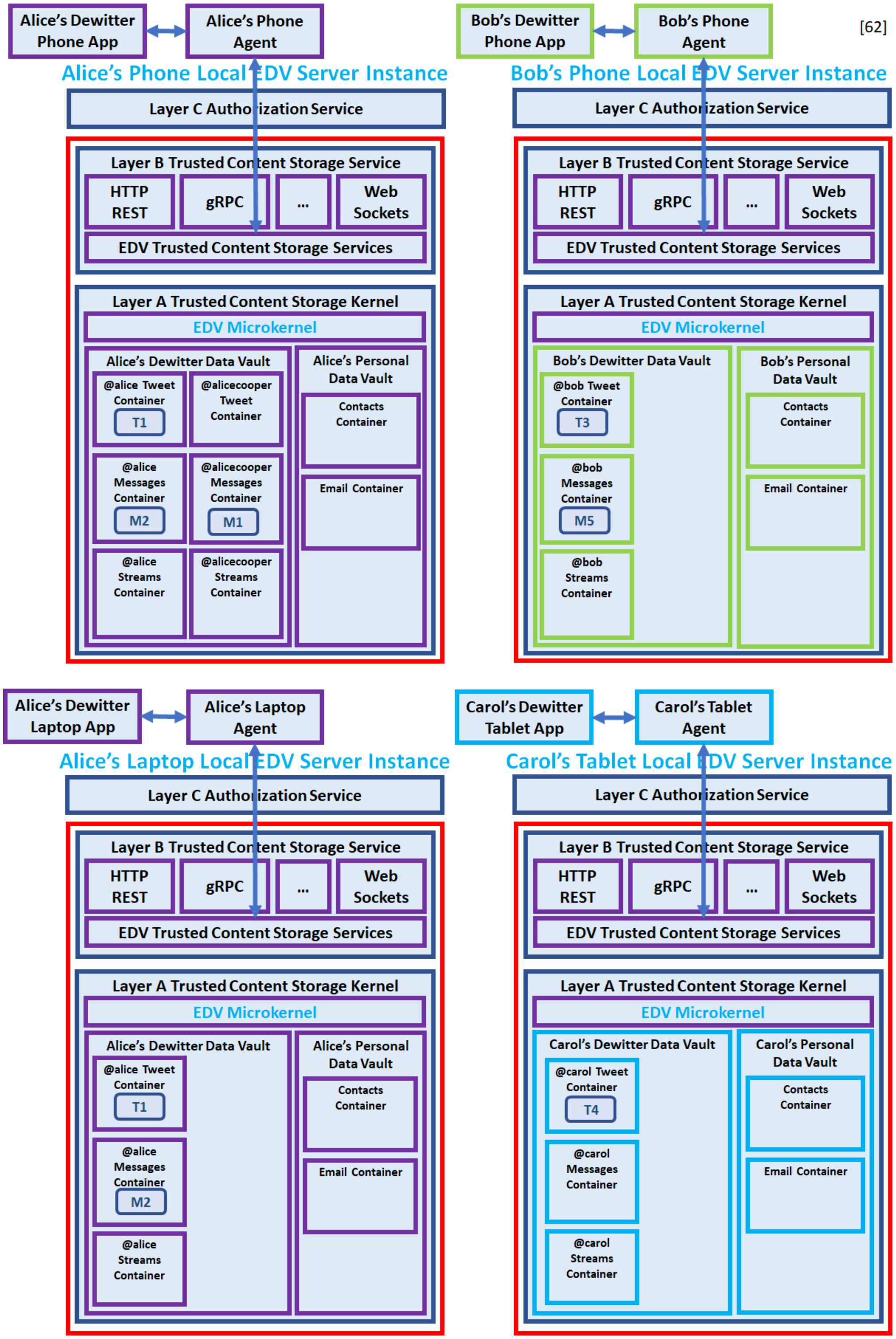
[3] Tianyin, Xu. et al. Cuckoo: Towards Decentralized, Socio-Aware Online Microblogging Services and Data Measurement, 2010.

4.2. Inactive	Inactive has an account (identity) and has created a few tweets but otherwise, hasn’t used Dewitter for several months.
4.3. Contributor	Contributor has one or more accounts (identities). Contributor likes to create new tweets a lot more than replying, retweeting, or searching. Contributor scans a lot of tweets (vs. reading tweets in-depth or diving into a thread of replies). Contributor uses reply, retweet, and searching infrequently. Contributor creates as well as re-uses a lot of #hashtags. Contributor uses a lot of photos and other image resources. Contributor comes in 3 variations based on the ratio of tweets created vs. tweets read: - Heavy Contributor (1 new tweet created for every 10 tweets read) - Medium Contributor (1 new tweet created for every 100 tweets read) - Light Contributor (1 new tweet created for every 1000 tweets read) Contributor comes also comes in 3 variations based on the frequency of use: Daily Contributor, Weekly Contributor, and Monthly Contributor.
4.4. Collaborator	Collaborator has one or more accounts (identities). Collaborator likes to create, read, and reply. Collaborator uses some photos and other image resources. Collaborator uses search. Collaborator uses reply a lot more than retweet. Collaborator comes in 3 variations: Daily Collaborator, Weekly Collaborator, and Monthly Collaborator.
4.5. Influencer	Influencer has one or more accounts (identities). Influencer primarily uses retweet and search – a lot more than any other feature. Influencer has millions of followers. Influencer has thousands of accounts it follows. Influencer is a daily user (Daily Influencer).
4.6. Promoter	Promoter has one or more accounts (identities). Promoter primarily creates new tweets as part of promotional campaigns. Promoter also uses retweet and search. Promoter has a few followers. Promoter has thousands of accounts it follows. Promoter is a daily user (Daily Promoter).
4.7. Newbie	Newbie doesn’t have a Dewitter account (identity) but is thinking of getting an account (identity).

NOTE: Later on in the app scenario, the actors may be assigned one of these roles.

Dewitter Platform Architecture Reference Model (Dewitter ARM)

The following reference architecture model illustrates the list of assumptions that follow the diagram.


Alice's Dewitter Phone App
Alice's Phone Agent
Alice's Phone Local EDV Server Instance
Layer C Authorization Service
Layer B Trusted Content Storage Service
HTTP REST
gRPC
...
Web Sockets
EDV Trusted Content Storage Services
Layer A Trusted Content Storage Kernel
EDV Microkernel
Alice's Dewitter Data Vault
@alice Tweet Container
T1
@alicecooper Tweet Container
@alice Messages Container
M2
@alicecooper Messages Container
M1
@alice Streams Container
@alicecooper Streams Container
Alice's Personal Data Vault
Contacts Container
Email Container
Bob's Dewitter Phone App
Bob's Phone Agent
Bob's Phone Local EDV Server Instance
Layer C Authorization Service
Layer B Trusted Content Storage Service
HTTP REST
gRPC
...
Web Sockets
EDV Trusted Content Storage Services
Layer A Trusted Content Storage Kernel
EDV Microkernel
Bob's Dewitter Data Vault
@bob Tweet Container
T3
@bob Messages Container
M5
@bob Streams Container
Bob's Personal Data Vault
Contacts Container
Email Container
Alice's Dewitter Laptop App
Alice's Laptop Agent
Alice's Laptop Local EDV Server Instance
Layer C Authorization Service
Layer B Trusted Content Storage Service
HTTP REST
gRPC
...
Web Sockets
EDV Trusted Content Storage Services
Layer A Trusted Content Storage Kernel
EDV Microkernel
Alice's Dewitter Data Vault
@alice Tweet Container
T1
@alice Messages Container
M2
@alice Streams Container
Alice's Personal Data Vault
Contacts Container
Email Container
Carol's Dewitter Tablet App
Carol's Tablet Agent
Carol's Tablet Local EDV Server Instance
Layer C Authorization Service
Layer B Trusted Content Storage Service
HTTP REST
gRPC
...
Web Sockets
EDV Trusted Content Storage Services
Layer A Trusted Content Storage Kernel
EDV Microkernel
Carol's Dewitter Data Vault
@carol Tweet Container
T4
@carol Messages Container
@carol Streams Container
Carol's Personal Data Vault
Contacts Container
Email Container

[62]

Figure 3. Dewitter Platform Architecture Reference Model (Dewitter ARM)

EDV Server Instances, EDV Data Vaults, and Containers

5. EDV Server Instances, EDV Data Vaults, and Containers
    a. Each actor has its own personal Local EDV Server Instance – one on each of their devices.
    b. For example, Alice has a separate EDV Server Instance on each of Alice's Phone and Alice's Laptop.
    c. Each actor's local server instance hosts multiple EDV Data Vaults.
    d. An EDV data vault has 1 or more Containers for storing different domains of resources.

Personal Data Vaults and Containers

6. Personal Data Vaults and Containers
    a. An actor can have 1 or more EDV Data Vaults attached to the personal Local EDV Server Instance running on each of the actor's devices.
    b. For example, Bob has 2 EDV Data Vaults: a Personal Data Vault and a Dewitter Data Vault.
    c. An EDV Data Vault can contain 1 or more Containers.
    d. A Container is used for storing content (i.e. a collection of resources)
    e. For example, Bob's Personal Data Vault (one his Phone, his only device) has 2 Containers: one for Bob's Contacts and the other for Bob's Email.
    f. For example, Alice's Dewitter Data Vault on Alice's Phone has64 Containers: 3 Containers for each of Alice's Dewitter accounts (identities): @alice and @alicecooper.
    g. For example, Alice's Dewitter Data Vault on Alice's Laptop has 3 Containers for Alice's @alice Dewitter account (identity):
        i. @alice Tweet Container (for locally created Tweets as well as Tweets received from other Person Agents),
        ii. @alice Messages Container (for locally created Direct Messages directed to others as well Direct Messages received from others), and
        iii. @alice Streams Container (for locally created Stream Items as well as Streams received from others).

Personal Agents

7. Personal Agents
    a. Each actor has a Personal Agent (a software client app) running on each of their devices.
    b. For example, Alice has Alice's Phone Agent and Alice's Laptop Agent.
    c. A Personal Agent is expected to automatically start operations when a device is turned on.
    d. A Personal Agent is expected to automatically resume operations when a device wakes up from Hibernation mode.
    e. Personal Agents can talk to each other (DIDComm?).
    f. An actor's Personal Agent on a device is the only client app that can interact with that actor's EDV Data Vaults attached to the actor's EDV Server Instance on that particular device.
    g. For example, Alice's Phone Agent is the only client app that can interact with Alice's Dewitter Data Vault and Alice's Personal Data Vault attached to Alice's Phone Local EDV Server Instance.

h. For example, Alice's Laptop is frequently in Hibernation mode (leading to an eventual Powered Off mode). When Alice's Laptop restarts, Alice's Laptop Agent automatically restarts.

Dewitter Data Vault

8. Dewitter Data Vault
    a. An actor's Dewitter Data Vault (on a particular device) has 1 or more Dewitter account (identity) Containers.
    b. For example, Alice's Dewitter Data Vault has 1 Tweet Container for each of Alice's Dewitter accounts (identities): @alice and @alicecooper.
    c. Dewitter Data Vault account (identity) Tweet Containers are used to store and manage tweets.
    d. For example, the @carol Tweet Container in Carol's Dewitter Data Vault on Alice's Tablet is used to store tweet T4.
    e. Similarly, Direct Messages are store in the @carol Messages Container and Stream Items (images, audio, and video) are stored in the @carol Streams Container.

SOLUTION CONCEPT

The solution concept for the Dewitter (Decentralized Twitter) App Scenario is described below. The solution concept variances for the Distwitter (Distributed Twitter) variation of the app scenario are described after the Dewitter App Scenario solution concept.

Dewitter Decentralized App Scenario

The Dewitter data model definitions, protocol operations, and use cases are described below.

Dewitter Data Model Definitions

This section describes how tweets are stored in Dewitter's fully decentralized architecture.

9. **Tweet Item**. A Tweet Item (Tweet) is a virtual base class for all of the different Dewitter Tweet Item Types. A Tweet Item is indexed and retrievable by a Tweet Key.
    a. **Tweet Item Type**. The type of a Tweet Item[4]. The types of Dewitter Tweet Items include:
        i. General Tweets
        ii. General Tweets with @Mentions
        iii. Replies
        iv. Retweets without Comments
        v. Retweets with Comments
        vi. News Tweets[5]
        vii. Promoted Tweets[6]
        viii. Direct Messages
    b. NOTE: Promoted Tweets[7] are ordinary Tweets purchased by advertisers who want to reach a wider group of users or spark engagement from their existing followers. All Promoted Tweets are clearly labeled as "Promoted" when an advertiser is paying for their placement on Twitter. In every other respect, Promoted Tweets are just like regular Tweets and can be retweeted, replied to, liked, and more.
10. **Tweet Key**. Each Tweet Item is identified by a key consisting of a pair of integer values:
    a. **DateTimeKey**: a 64-bit integer value representing the time-of-day in Ticks (one hundred nanoseconds or one ten-millionth of a second).
    b. **DiscriminatorKey**: a 128-bit binary value representing a specific random GUID (16 bytes)
    c. A collection of Tweet Keys (and in turn, Tweet Items) are intended to be ordered and sortable with the DateTimeKey value as the primary key, and the DiscriminatorKey value as the secondary key.
    d. NOTE: Alternatively, if each account (identity) is given a unique (possibly sequential) 64-bit integer identifier value when they register to use Dewitter, this value could be used for the DiscriminatorKey value.

[4] Twitter. *About different types of Tweets* (https://help.twitter.com/en/using-twitter/types-of-tweets)

[5] News Tweets are considered out of scope for this version of the Dewitter app scenario.

[6] Promoted Tweets are considered out of scope for this version of the Dewitter app scenario.

[7] Twitter. *What are Promoted Tweets?* (https://business.twitter.com/en/help/overview/what-are-promoted-tweets.html)

11. **Tweet Key Range**. A Tweet Key Range is a pair of Tweet Keys (called **StartKey** and **EndKey**) that specify the starting and ending Tweet Keys for an ordered or unordered collection of tweets.
    a. DateTime.MinValue and DateTime.MaxValue are valid DateTimeKey values to use a Tweet Key Range as a search parameter.
    b. Int64.MinValue and Int64.MinValue are valid DiscriminatorKey values to use a Tweet Key Range as a search parameter.
    c. NOTE: It's indeterminate whether a particular collection of tweets is contiguous (i.e. with no missing tweets) except for the single case of a collection of all the Tweets created by a local account (identity).
12. **Tweet Notifications**. Tweet Notifications are lightweight objects consisting of little more than a Tweet Key for an existing Tweet Item (plus a little bit of metadata). They are intended to notify the receiving Personal Agent that one of the receiving Personal Agent's specific Following has a new, changed, or tombstoned Tweet.
13. **Tweet Query**. A Tweet Query is composed of a Tweet Key Range (called **Tweet Query Range**) plus a Tweet Query Filter.
    a. **Tweet Query Filter**. A Tweet Query Filter is a filter specifying additional query filtering parameters such as:
        i. For a specific person who created the Tweet,
        ii. Date and time that the Tweet was created,
        iii. Tweet Item Type,
        iv. Tweets containing one or more matches from a list of #Hashtags,
        v. Tweets containing one or more matches from a list of @Mentions,
        vi. Tweets containing one or more matches from a list of search words, etc.
14. **Tweet Like Counter**. A Tweet Like Counter is a property associated with each Tweet Item that records the number of unique accounts (identities) that have "liked" the Tweet Item.
    NOTE: It is the "number of unique accounts (identities)". This implies that there also needs to be a list of unique accounts (identities) that have liked a Tweet Item associated with that item.
    a. **Tweet Like Account List**. A Tweet Like Account List is a list of unique accounts (identities) that have liked the Tweet Item.
    b. NOTE: The Tweet Like Account List is not implemented in this version of the Dewitter app scenario.
15. **Stream Item**. A Stream Item (Stream) is a virtual base class for all of the different Dewitter Stream Item Types. A Stream Item is indexed and retrievable by a Stream Key.
    a. **Stream Item Type**. The type of a Stream Item. The types of Dewitter Stream Items include:
        i. Image
        ii. Audio
        iii. Video
16. **Stream Key**. A Stream Key is a Tweet Key (for this version of the app scenario).
17. **Stream Query**. A Stream Query is composed of a Stream Query Filter.
    a. **Stream Query Filter**. A Stream Query Filter is a filter specifying query filtering parameters such as:
        i. For a specific person who created the Stream,
        ii. Date and time that the Stream was created,
        iii. Stream Item Type

18. Tweet Items and Tweet Notifications are stored in Dewitter Tweet Containers in personal Dewitter Data Vaults.
    a. All of the Tweets (include all of the different types of Tweet Items) created by an account (identity) are stored in the same Tweet Container named for the account (identity).
    b. All of the (recent) Tweets (include all of the different types of Tweet Items) for all of the account's (identity's) Following are also stored in the account's (identity's) Tweet Container when/as they are received.
19. Reserved
20. Reserved
21. Reserved
22. Reserved

Personal Agent to Local EDV Server Instance Communications

23. Because of the fully decentralized focus of the Dewitter platform, Alice's personal Agent and Alice's Local EDV Server Instance are naturally co-located (deployed) on the same device (e.g. either Alice's smartphone or laptop) as illustrated in the diagram below.
    a. Alice's Agent talks directly to Alice's Local EDV Server Instance directly via the Layer B EDV Trusted Content Storage Services API – not via one of the Layer B Trusted Content Storage Service remote access service endpoints (e.g. HTTP).
    b. The Layer B EDV Trusted Content Storage Services, in turn, talks directly to the Layer A Trusted Content Storage Kernel; which, in turn, talks directly to the EDV Microkernel.
    c. The EDV Microkernel manages all access and operations against each of the attached EDV Data Vaults.

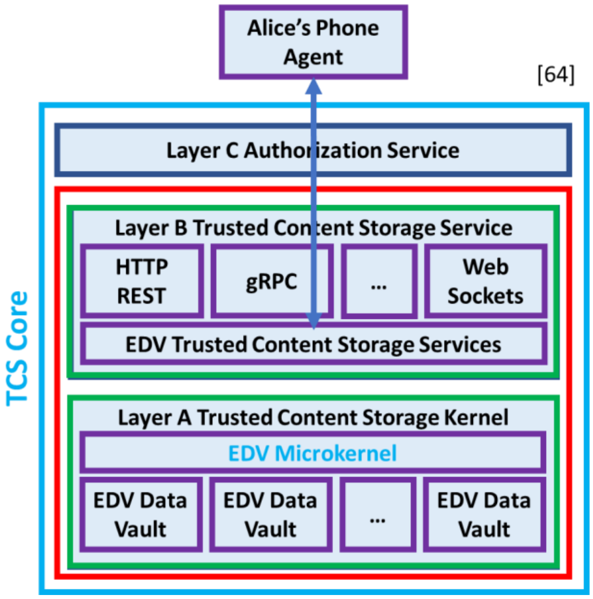

Alice's Phone Agent
[64]
Layer C Authorization Service
Layer B Trusted Content Storage Service
HTTP REST
gRPC
...
Web Sockets
EDV Trusted Content Storage Services
TCS Core
Layer A Trusted Content Storage Kernel
EDV Microkernel
EDV Data Vault
EDV Data Vault
...
EDV Data Vault

Figure 4. Personal Agent to Local EDV Server Instance Communications

Dewitter Personal Agent Protocol Operations

This section describes how basic CRUD operations are performed on Tweets, Direct Messages, and Streams stored in Dewitter's fully decentralized architecture. The types of operations that need to be supported by the Dewitter Protocols include the following.

Create/Update a Tweet Item or Stream Item in a Personal Local EDV Service Instance

24. Create/Update a Tweet Item (or Streams Item) in a Dewitter Tweet Container (or Dewitter Messages Container or Dewitter Streams Container) in a Personal Local EDV Service Instance
    a. Ability to create a new Tweet Item (or Streams Item) in the personal Dewitter Tweet Container (or Dewitter Messages Container or Dewitter Streams Container) in a personal Local EDV Server Instance
    b. Ability, given a Tweet Key (or Stream Key), to update an existing Tweet Item (or Streams Item) in the personal Dewitter Tweet Container (or Dewitter Messages Container or Dewitter Streams Container) in a personal Local EDV Server Instance
    c. Ability, given a Tweet Key (or Stream Key), to tombstone an existing Tweet Item (or Streams Item) in the personal Dewitter Tweet Container (or Dewitter Messages Container or Dewitter Streams Container) in a personal Local EDV Server Instance

Query a Personal Local EDV Service Instance for a List of Tweet Keys or a Collection of Tweet Items

25. Query a Personal Local EDV Service Instance for a List of Tweet Keys or a Collection of Tweet Items
    a. Ability to query a personal Local EDV Server Instance for and return a collection of Tweet Keys given a Tweet Query parameter
    b. Ability to query a personal Local EDV Server Instance for and return a collection of Tweet Items given a Tweet Query parameter

Query Another Personal Agent or List of Personal Agents for a List of Tweet Keys

26. Query Another Personal Agent or List of Personal Agents for a List of Tweet Keys
    a. Ability to query another Personal Agent for and return a list of Tweet Keys given a Tweet Query as a parameter
    b. Ability to query a list of Personal Agents for and return a list of Tweet Keys given a Tweet Query as a parameter
    c. Ability to query a list of Personal Agents for and return a list of Tweet Keys given a corresponding list of Tweet Key Queries as a parameter

Query Another Personal Agent or List of Personal Agents for a Collection of Tweet Items

27. Query Another Personal Agent or List of Personal Agents for a Collection of Tweet Items
    a. Ability to query another Personal Agent for and return a collection of Tweet Items given a Tweet Query as a parameter
    b. Ability to query a list of Personal Agents for and return a collection of Tweet Items given a Tweet Query as a parameter
    c. Ability to query a list of Personal Agents for and return a collection of Tweet Items given a corresponding list of Tweet Key Ranges as a parameter

Block a Query from Another Personal Agent

28. Block a Query from Another Personal Agent
    a. A receiving Personal Agent can totally block the receipt and acceptance of a Tweet Query from a particular Person Agent.

Query a Personal Local EDV Service Instance for a Stream Item

29. Query a Personal Local EDV Service Instance for a Stream Item
    a. Ability to query a personal Local EDV Server Instance for and return Stream Key given Stream Query parameter
    b. Ability to retrieve and accept (download) a Stream Item from a personal Local EDV Service Instance given its Stream Key following the Generic Replication Pipeline pattern

Query Another Personal Agent or List of Personal Agents for a Stream Item

30. Query Another Personal Agent or List of Personal Agents for a Stream Item
    a. Ability to query another Personal Agent for and return a Stream Key given a Stream Query parameter
    b. Ability to retrieve and accept (download) a Stream Item from another Personal Agent given its Stream Key following the Generic Replication Pipeline pattern

Send/Receive/Accept/Block Single Tweet Item to Another Personal Agent

31. Send/Receive/Accept/Block Single Tweet Item to Another Personal Agent
    a. Ability, given a Tweet Key, for a Personal Agent to send a Tweet Item to another personal Agent
    b. Ability, given a Tweet Key, for a Personal Agent receive (and, optionally, accept) a Tweet Item sent from another personal Agent
    c. A receiving Personal Agent can totally block the receipt and acceptance of a Tweet Item from a particular Person Agent.

Send/Receive/Accept/Block Batch of Tweet Items to Another Personal Agent

32. Send/Receive/Accept Batch of Tweet Items to Another Personal Agent
    a. Ability, given a list of Tweet Keys, for a Personal Agent to send a batch of Tweet Items to another personal Agent
    b. Ability, given a list of Tweet Keys, for a Personal Agent receive (and, optionally, individually accept) a batch of Tweet Items sent from another personal Agent
    c. A receiving Personal Agent can totally block the receipt and acceptance of Tweet Items from a particular Person Agent.

Broadcast/Receive/Accept/Block a Batch of Tweet Items to Multiple Personal Agents

33. Broadcast a Batch of Tweet Items to Multiple Personal Agents
    a. Ability, given a list of Tweet Keys and a list of other Personal Agents, for a Personal Agent to broadcast one or more Tweet Items (a batch of Tweet Items) to list of Personal Agents to be received and, optionally, accepted.
    b. A receiving Personal Agent can totally block the receipt and acceptance of Tweet Items from a particular Person Agent.

Send/Broadcast/Receive/Accept/Block a Batch of Tweet Notifications to Multiple Personal Agents

34. Send/Broadcast/Receive/Accept/Block a Batch of Tweet Notifications to Multiple Personal Agents
    a. Ability, given a list of Tweet Keys and a list of other Personal Agents, for a Personal Agent to broadcast one or more Tweet Notifications (a batch of Tweet Notifications) to list of Personal Agents to be received and, optionally, accepted.
    b. A receiving Personal Agent can totally block the receipt and acceptance of Tweet Notifications from a particular Person Agent.

Update a Tweet Item in Another Personal Local EDV Service Instance

35. Create/Update a Tweet Item in a Dewitter Tweet Container (or Dewitter Messages Container) in a Personal Local EDV Service Instance
    a. Ability, given a Tweet Key, to update an existing Tweet Item in the personal Dewitter Tweet Container (or Dewitter Messages Container) in a personal Local EDV Server Instance

Dewitter Use Cases

Given the above assumptions, principles, requirements, and other considerations, this section describes how the most common Twitter operations can be implemented on the Dewitter platform.

36. Create ("Tweet")
    a. Using protocol operation 24a (create), create a new General Tweet in the account's (identity's) Dewitter Tweet Container in the local, personal Dewitter Data Vault attached to the device's personal Local EDV Server Instance.
37. Notify Your Followers
    a. Using protocol operation 30a (notification broadcast), send a Tweet Notification for each Tweet in a list of new, changed, or tombstoned Tweets to a list of Personal Agents (usually Following).
38. Follow Your Following
    a. Using protocol operation 26b (Tweet Keys query), 26c (Tweet Keys batch query), 27b (Tweet Items query), or 27c (Tweet Items batch query), query the Dewitter Tweet Container(s) in the personal Dewitter Data Vault attached to the personal Local EDV Server Instance for each of your Following with a Tweet Query StartKey set to the Tweet Key of the person's last (highest) reported Tweet Item's Tweet Key and the EndKey set equal to MaxValues.
39. Read a Specific Tweet
    a. Read a Specific Tweet from personal Local EDV Server Instance: Using protocol operation 24b (Tweet Items query), query the Dewitter Tweet Container(s) in the Dewitter Data Vault attached to your personal Local EDV Server Instance for a particular Tweet Key
    b. Read a Specific Tweet from another Personal Agent: Using protocol operation 27a (Tweet Items query), query the Dewitter Tweet Container(s) in the Dewitter Data Vault attached to the Personal EDV Server Instance connected to the other Personal Agent for a particular Tweet Key
40. Reply
    a. Read the Parent Tweet using operation 39 (Read a Specific Tweet).

   b. Using protocol operation 24a (create), create a new Reply Tweet in the account's (identity's) Dewitter Tweet Container attached to the device's personal Local EDV Server Instance – linking the Reply Tweet to the parent Tweet Item.

41. Retweet (RT) without Comment
    a. Read the Parent Tweet using operation 39 (Read a Specific Tweet).
    b. Using protocol operation 24a (create), create a new Retweet Tweet without Comment in the account's (identity's) Dewitter Tweet Container attached to the device's personal Local EDV Server Instance – linking the Retweet Tweet to the parent Tweet Item
42. Retweet (RT) with Comment
    a. Read the Parent Tweet using operation 39 (Read a Specific Tweet).
    b. Using protocol operation 24a (create), create a new Retweet Tweet with Comment in the account's (identity's) Dewitter Tweet Container attached to the device's personal Local EDV Server Instance – linking the Retweet Tweet to the parent Tweet Item.
43. Direct Message (DM)
    a. Using protocol operation 24a (create), create a new Direct Message Tweet in the account's (identity's) Dewitter Messages Container in the local, personal Dewitter Data Vault attached to the device's personal Local EDV Server Instance
44. Reply to a Direct Message (DM)
    a. Using protocol operation 24a (create), create a new Direct Message Tweet in the account's (identity's) Dewitter Messages Container in the local, personal Dewitter Data Vault attached to the device's personal Local EDV Server Instance – linking the Direct Message Tweet to the parent Direct Message Tweet.
45. Search Local Personal Dewitter Data Vault
    a. Search all Tweets in the Dewitter Tweet Container (or Dewitter Messages Container) in the local, personal Dewitter Data Vault returning a list of Tweet Keys: using protocol operation 25a, query the Dewitter Tweet Container(s) in the local, personal Dewitter Data Vault attached to the local, personal Local EDV Server Instance with a Tweet Query StartKey set equal to MinValues and the EndKey set equal to MaxValues. An option Tweet Query Filter can also be used (see below).
    b. Search all Tweets in the Dewitter Tweet Container (or Dewitter Messages Container) in the local, personal Dewitter Data Vault within a certain DateTime range returning a list of Tweet Keys: using protocol operation 25a, query the Dewitter Tweet Container(s) in the local, personal Dewitter Data Vault attached to the local, personal Local EDV Server Instance with a Tweet Query StartKey.DateTimeKey set equal to the starting time, StartKey.DiscrimatorValue set equal to MinValue, EndKey.DateTimeKey set equal to the ending time, and EndKey.DiscriminatorValue set equal to MaxValue.
    c. Search all Tweets in the Dewitter Tweet Container (or Dewitter Messages Container) in the local, personal Dewitter Data Vault containing a specific list of #Hashtags return a collection of Tweet Items: using protocol operation 25a, query the Dewitter Tweet Container(s) in the local, personal Dewitter Data Vault attached to the local, personal Local EDV Server Instance with a Tweet Query Filter set equal to the list of #HashTag values.
    d. Search all Tweets in the Dewitter Tweet Container (or Dewitter Messages Container) in the local, personal Dewitter Data Vault containing a specific list of @Mentions returning

a collection of Tweet Items: using protocol operation 25a, query the Dewitter Tweet Container(s) in the local, personal Dewitter Data Vault attached to the local, personal Local EDV Server Instance with a Tweet Query Filter set equal to the list of @Mention values.

46. Search Personal Dewitter Data Vault connected to Another Personal Agent
    a. Search all Tweets in the Dewitter Tweet Container (or Dewitter Messages Container) in the personal Dewitter Data Vault connected to another Personal Agent returning a collection of Tweet Items: using protocol operation 27a, query the Dewitter Tweet Container(s) in the personal Dewitter Data Vault attached to the personal Local EDV Server Instance with an optional Tweet Query.
47. Photos and Image Resources
    a. Query for and return an image stream used in a Tweet from the personal Dewitter Data Vault connected to another Person Agent: using protocol 30a (stream query), query the Dewitter Stream Container(s) in the personal Dewitter Data Vault attached to the personal Local EDV Server Instance with a given Stream Query.
48. Like a Tweet
    a. Read the Tweet to be Liked using operation 39 (Read a Specific Tweet).
    b. Increment the Tweet's Like Counter by 1
    c. Using protocol operation 35a, update the Tweet's Like Counter in the Dewitter Tweet Container in the personal Dewitter Data Vault attached to the personal Local EDV Server Instance.
    d. NOTE: The "increment and update the Tweet's Like Counter" needs to be an atomic operation executed remotely while possibly multiple Personal Agents are also trying to increment the same Tweet's Like Counter at the same time.
    e. NOTE: Like a Tweet is very difficult to do in a decentralized system – almost impossible – without the ability to define, deploy, and remotely execute a Layer A Trusted Content Storage Kernel-level stored procedure (or the equivalent) to perform the Like counter increment operation directly on a locked resource in EDV Data Vault.
    f. NOTE: The Tweet Like Account List is not implemented in this version of the Dewitter app scenario.
49. Blocking
    a. Using protocol operation 28a, the Dewitter app can inform its Personal Agent not to receive (let alone accept) any messages from another Personal Agent.
    b. To learn more about Blocking, see APPENDIX A.
50. Muting
    a. The Dewitter app (and in turn, its Personal Agent) should accept and receive all messages from all non-blocked Personal Agents (assuming the agent is only servicing the Dewitter app in this app scenario). It is a UX concern to mute the display of Tweets from muted Personal Agents for specific accounts (identities).
    b. Option 2: Using protocol operation 28a, the Dewitter app can inform its Personal Agent not to receive (let alone accept) any messages from another Personal Agent. (But this isn't really in the spirit of what Muting is supposed to do. See APPENDIX A.)

51. Notifications
    c. Notifications are not addressed adequately in this version of the Dewitter app scenario. All of the current Dewitter functionality is based primarily on a Request/Send/Receive/Accept pull-then-push model. To adequately address this use case, TCS Layer D Replication Services would be used to implement a pub-sub model where an inbound replication connection would be created for each Following account (identity) and corresponding outbound replication connections would be created for each Follower.

Notes

1. Offline Personal Agents. This is outside the scope of this app scenario. It is assumed that a Personal Agent will queue outbound messages for another Personal Agent if the second agent was offline.
2. Offline personal Local EDV Server Instances. This is outside the scope of this app scenario. It is assumed that a Personal Agent will queue outbound messages for an offline personal Local EDV Server Instance.
3. Data Metrics and Analysis. Data Metrics and Analysis require the logging and analysis of extensive, detailed information gathered from each of the above operations – in addition to static analysis of a local Personal Dewitter Data Vault. Note that mass aggregation of static data across all accounts (identities) is no longer possible unless it is provided for in the Dewitter API. An analysis is only available for an account's (identity's) local Neighborhood.
4. Synchronization of Account (Identity) Containers across Dewitter Data Vaults attached to different Local EDV Server Instances. In Alice's world where Alice has 2 devices and one of the two of Alice's accounts (identities) has been configured on both devices (e.g. @alice), a TCS Layer 4 Replication Service is expected to be used with 2 Replication Connections to support bi-directional synchronization. See APPENDIX B.

Distwitter Distributed App Scenario

The Distwitter Distributed Twitter app scenario is an extension to the Dewitter Decentralized Twitter where centralized servers and server-hosted EDV Server Instances are allowed.

Centralized Servers

1. Centralized (and possibly geo-replicated) servers are used to host EDV Server Instances (and server-hosted EDV data vaults are attached to them).

Server-hosted EDV Server Instances

1. Server-hosted EDV Server Instances run on centralized servers.
2. Server-hosted EDV Data Vaults are stored locally on the server (cluster) (or Network Attached Storage (NAS) – also possibly geo-replicated).
3. Server-hosted EDV Data Vaults are attached to a EDV Server Instance running on the same server (cluster).

Notes

1. In the Distwitter Distributed Twitter app scenario, a large number of performance and network communication optimizations are possible but only with significantly reduced decentralization benefits. For example, centralization of a master Tweet database (still built using EDV Data Vaults using many of the data model and protocol operations patterns described above).

CONCLUSION

If I have seen further, it is by standing on the shoulders of Giants.

[Issac Newton, 1675]

The purpose of this document is to describe a Decentralized Twitter (Dewitter) App Scenario based on the Trusted Content Storage Architecture (TCS Stack).

The next steps are to proceed with a prototype implementation of a proof-of-concept of all the features described in this document.

Acknowledgments

The responsibility for any errors or omissions lies with me alone.

APPENDICES

APPENDIX A – RESOURCES

Twitter Help Documentation

1. Using Twitter: https://help.twitter.com/en/using-twitter
2. Types of Tweets: https://help.twitter.com/en/using-twitter/types-of-tweets
3. Authorization use cases: https://help.twitter.com/en/using-twitter/twitter-conversations
4. Liking Tweets: https://help.twitter.com/en/using-twitter/liking-tweets-and-moments
5. Notifications: https://help.twitter.com/en/managing-your-account/notifications-on-mobile-devices
6. Policies: https://help.twitter.com/en/rules-and-policies/twitter-limits, https://help.twitter.com/en/using-twitter/mentions-and-replies
7. Glossary: https://help.twitter.com/en/glossary
8. Blocking and Muting: https://help.twitter.com/en/using-twitter#blocking-and-muting
9. Promoted Tweets: https://business.twitter.com/en/help/overview/what-are-promoted-tweets.html

Blocking

Credit: https://help.twitter.com/en/using-twitter/blocking-and-unblocking-accounts

Block is a feature that helps you control how you interact with other accounts on Twitter. This feature helps users in restricting specific accounts from contacting them, seeing their Tweets, and following them.

Note: Learn about our advanced block options for more detailed information on this feature.

Some important things to know about block:

- Accounts you have blocked cannot follow you, and you cannot follow an account you have blocked.
- Blocking an account you are currently following will cause you to unfollow that account (and them to unfollow you). If you decide to unblock that account, you will have to follow that account again.
- Blocked accounts do not receive a notification alerting them that their account has been blocked. However, if a blocked account visits the profile of an account that has blocked them, they will see they have been blocked (unlike mute, which is invisible to muted accounts).
- If you block an account and they choose to report your account, any of your Tweets that directly mention them will be available for them to view and attach during the reporting process.
- You will not receive notifications from accounts you block, or accounts that you do not follow who mention you in conversations started by accounts that you block. You will, however, see notifications from accounts you follow when they mention you in a conversation started by an account you block. If you'd like to view all of your mentions, you can do so by searching your username.

Important: Blocking only works if the account you've blocked is logged in on Twitter. For example, if the account you've blocked isn't logged in or is accessing Twitter content via a third party, they may be able to see your public Tweets. Please keep this in mind when you choose to share content on Twitter.

Blocked accounts cannot:

- Follow you
- View your Tweets when logged in on Twitter (unless they report you, and your Tweets mention them)
- Find your Tweets in search when logged in on Twitter
- Send Direct Messages to you
- View your following or followers lists, likes or lists when logged in on Twitter
- View a Moment you've created when logged in on Twitter
- Add your Twitter account to their lists
- Tag you in a photo

Tweets from blocked accounts will not appear in your timeline. However, please note that you may see Tweets or notifications in your timeline for the following:

- Tweets from others you follow that mention accounts you have blocked.
- Tweets that mention you, along with an account you have blocked.

Muting

Credit: https://help.twitter.com/en/using-twitter/twitter-mute

Mute is a feature that allows you to remove an account's Tweets from your timeline without unfollowing or blocking that account. Muted accounts will not know that you’ve muted them and you can unmute them at any time. To access a list of accounts you have muted, visit your muted accounts settings on twitter.com or your app settings on Twitter for iOS or Android.

To learn about muting notifications, read about our advanced muting options on Twitter.

Some important things to know about mute:

- Muted accounts can follow you and you can follow muted accounts. Muting an account will not cause you to unfollow them.
- Muting an account does not impact the account’s ability to send you a Direct Message.
- You will no longer receive push or SMS notifications from any muted account.

For muted accounts that you follow:

- Replies and mentions by the muted account will still appear in your Notifications tab.
- Tweets from a muted account – posted before the account was muted – will be removed from your Home timeline.
- When you click or tap into a conversation, replies from muted accounts will be visible.

For muted accounts that you do not follow:

- Replies and mentions will not appear in your Notifications tab.
- If you mute an account you do not follow and they start a conversation that mentions you, you will only receive notifications from those you do follow who reply in the conversation and mention you. If you’d like to view all of your mentions, you can do so by searching your username.
- When you click or tap into a conversation, replies from muted accounts will be not visible.
- If an account you have not muted Reweets with comments an account you have muted, the Tweet will be hidden with a This Tweet is unavailable message.

Mute is a feature that allows you to remove an account's Tweets from your timeline without unfollowing or blocking that account. Muted accounts will not know that you’ve muted them and you can unmute them at any time. To access a list of accounts you have muted, visit your muted accounts settings on twitter.com or your app settings on Twitter for iOS or Android.

To learn about muting notifications, read about our advanced muting options on Twitter.

Some important things to know about mute:

- Muted accounts can follow you and you can follow muted accounts. Muting an account will not cause you to unfollow them.
- Muting an account does not impact the account’s ability to send you a Direct Message.
- You will no longer receive push or SMS notifications from any muted account.

For muted accounts that you follow:

- Replies and mentions by the muted account will still appear in your Notifications tab.
- Tweets from a muted account – posted before the account was muted – will be removed from your Home timeline.
- When you click or tap into a conversation, replies from muted accounts will be visible.

For muted accounts that you do not follow:

- Replies and mentions will not appear in your Notifications tab.
- If you mute an account you do not follow and they start a conversation that mentions you, you will only receive notifications from those you do follow who reply in the conversation and mention you. If you'd like to view all of your mentions, you can do so by searching your username.
- When you click or tap into a conversation, replies from muted accounts will be not visible.
- If an account you have not muted Reweets with comments an account you have muted, the Tweet will be hidden with a This Tweet is unavailable message.

Mute is a feature that allows you to remove an account's Tweets from your timeline without unfollowing or blocking that account. Muted accounts will not know that you've muted them and you can unmute them at any time. To access a list of accounts you have muted, visit your muted accounts settings on twitter.com or your app settings on Twitter for iOS or Android.

To learn about muting notifications, read about our advanced muting options on Twitter.

Some important things to know about mute:

- Muted accounts can follow you and you can follow muted accounts. Muting an account will not cause you to unfollow them.
- Muting an account does not impact the account's ability to send you a Direct Message.
- You will no longer receive push or SMS notifications from any muted account.

For muted accounts that you follow:

- Replies and mentions by the muted account will still appear in your Notifications tab.
- Tweets from a muted account – posted before the account was muted – will be removed from your Home timeline.
- When you click or tap into a conversation, replies from muted accounts will be visible.

For muted accounts that you do not follow:

- Replies and mentions will not appear in your Notifications tab.
- If you mute an account you do not follow and they start a conversation that mentions you, you will only receive notifications from those you do follow who reply in the conversation and mention you. If you'd like to view all of your mentions, you can do so by searching your username.
- When you click or tap into a conversation, replies from muted accounts will be not visible.
- If an account you have not muted Reweets with comments an account you have muted, the Tweet will be hidden with a This Tweet is unavailable message.

APPENDIX B – A GENERIC REPLICATION PIPELINE

A generic Replication Pipeline is illustrated below.

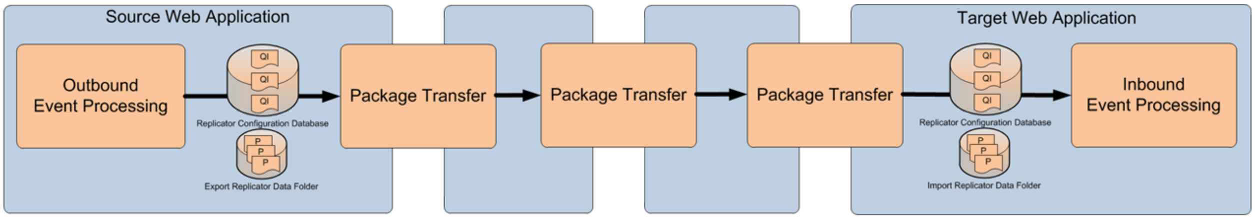

Source Web Application
Outbound Event Processing
QI
QI
QI
Replicator Configuration Database
P
P
P
Export Replicator Data Folder
Package Transfer
Package Transfer
Package Transfer
Target Web Application
QI
QI
QI
Replicator Configuration Database
P
P
P
Import Replicator Data Folder
Inbound Event Processing

Figure 5. A Generic Replication Pipeline (Credit: Syntergy Replicator for SharePoint)

Outbound Processing Service

Outbound Processing Service is responsible for capturing and recording Replication Events that occur in the Source Repository. Outbound Processing Service is controlled by Replication Maps which determine what Events need to be captured, packaged, and transferred to the Target Repository. Groups of Replication Events are packaged into two types of messages or objects: Queued Items and Replication Packages.

Queued Item

A Queued Item is a unit of work to be transferred to a Target Repository for remote execution. The Replication Web Service on a Target Repository is called to push a Queued Item from the Source Repository to a Target Repository.

Replication Package

A Replication Package is a collection of one or more Replication Events plus data about the changed information that is packaged in a format specific to the Replication Transport being used. When an Event is being processed, Outbound Processing Service calls the Source repository object model to extract the changed information from the Source Repository.

Package Transfer Service

The Package Transfer Service activity is responsible for the transfer and receipt of Queued Items and the downloading Replication Packages (Packages) from the Source Repository to the Target Repository. Package Transfer is the process that sits between Outbound Processing Service (on the Source Repository) and Inbound Processing Service (on the Target Repository).

Inbound Processing Service

Inbound Processing Service is responsible for processing the Queued Items and Packages received and accepted by the Target Repository and applying them to the repository. The Queued Items and Packages are applied to the Target Repository content base by calling the Target Repository object model.

APPENDIX C – LICENSES

MIT License

MIT License

Copyright (c) 2019-2021 Michael Herman (Toronto/Calgary/Seattle)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Creative Commons Attribution-ShareAlike 4.0 International Public License

Creative Commons Attribution-ShareAlike 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-ShareAlike 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

a. **Adapted Material** means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

b. **Adapter's License** means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

c. **BY-SA Compatible License** means a license listed at creativecommons.org/compatiblelicenses, approved by Creative Commons as essentially the equivalent of this Public License.

d. **Copyright and Similar Rights** means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

e. **Effective Technological Measures** means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

f. **Exceptions and Limitations** means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

g. **License Elements** means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution and ShareAlike.

h. **Licensed Material** means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

i. **Licensed Rights** means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

j. **Licensor** means the individual(s) or entity(ies) granting rights under this Public License.

k. **Share** means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

l. **Sui Generis Database Rights** means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

m. **You** means the individual or entity exercising the Licensed Rights under this Public License. **Your** has a corresponding meaning.

Section 2 – Scope.

a. **License grant**.

1. Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

   A. reproduce and Share the Licensed Material, in whole or in part; and

   B. produce, reproduce, and Share Adapted Material.

2. Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

3. Term. The term of this Public License is specified in Section 6(a).

4. Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

5. Downstream recipients.

   A. Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

   B. Additional offer from the Licensor – Adapted Material. Every recipient of Adapted Material from You automatically receives an offer from the Licensor to exercise the Licensed Rights in the Adapted Material under the conditions of the Adapter's License You apply.

C. No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

6. No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).

b. **Other rights**.

1. Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.

2. Patent and trademark rights are not licensed under this Public License.

3. To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.

Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

a. **Attribution**.

1. If You Share the Licensed Material (including in modified form), You must:

A. retain the following if it is supplied by the Licensor with the Licensed Material:

i. identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);

ii. a copyright notice;

iii. a notice that refers to this Public License;

iv. a notice that refers to the disclaimer of warranties;

v. a URI or hyperlink to the Licensed Material to the extent reasonably practicable;

B. indicate if You modified the Licensed Material and retain an indication of any previous modifications; and

C. indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.

2. You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.

3. If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.

b. **ShareAlike**.

In addition to the conditions in Section 3(a), if You Share Adapted Material You produce, the following conditions also apply.

1. The Adapter's License You apply must be a Creative Commons license with the same License Elements, this version or later, or a BY-SA Compatible License.

2. You must include the text of, or the URI or hyperlink to, the Adapter's License You apply. You may satisfy this condition in any reasonable manner based on the medium, means, and context in which You Share Adapted Material.

3. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply.

Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

a. for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

b. if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material, including for purposes of Section 3(b); and

c. You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

a. **Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.**

b. **To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.**

c. The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

a. This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

b. Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

   1. automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

   2. upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

c. For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

d. Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

a. The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

b. Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

Credit: https://creativecommons.org/licenses/by-sa/4.0/legalcode