**Description:** We are building an integration between the Signalmash platform and a company called Volt (textvolt.com). This integration is to designed to allow users to stream their SMS Webhook Events to Volt for Record Summarization and Reporting.

**Signalmash / Volt Insights Integration Proposal**

This is a proposal for an integration between Signalmash and Volt for the Insights product. Feel free to add comments to this document.

User Steps

1. A Signalmash user clicks an initial set-up button link inside Signalmash panel to that sends a request to Volt.
2. Volt triggers email with password setting option
3. Volt sends a request to Signalmash identifying the user’s account as a Volt user.
4. Signalmash begins sending messaging-related webhooks to Volt.
5. Signalmash users can login to Volt Insights via SSO using encoded link below (trusted by Volt) or directly to Volt Insights web application using username (same as Signalmash username) and password set in step 2.

Step 1: A Signalmash user clicks an outbound link to Volt.

The URL should include this information:

* Signalmash account ID — to be associated with a Volt organization
* Signalmash user ID — to be associated with a Volt user
* Cryptographic Signature — for verification by Volt that the link originated from Signalmash

Link URL Scheme

https://insights.textvolt.com/login/signalmash?account\_id=**<account\_id>**&user\_id=**<user\_id>**&signature=**<signature>**

* **<account\_id>** — Signalmash account ID
* **<user\_id>** — Signalmash user ID – this will be the Signalmash username (email)
* **<signature>** — a cryptographic digital signature using the elliptic curve digital signature algorithm (described under the heading Cryptographic Signature Scheme)

Cryptographic Signature Scheme (Elliptic Curve Digital Signature Algorithm)

Demo: [Link to a reference implementation](https://go.dev/play/p/XQP2kyMFj1N)

1. (Offline) Signalmash should generate an ECDSA public-private key pair and either publish the public key or communicate it to Volt.
2. The initial message should have this structure:
account\_id:**<account\_id>**|user\_id:**<user\_id>**
3. The initial message should be hashed using SHA-256 to generate a digest.
4. The SHA-256 digest should be signed with Signalmash’s private key using ECDSA, returning the calculated signature (a pair of integers denoted {r, s}).
5. The calculated signature should be ASN.1 encoded and returned in hexadecimal representation.

General workflow that will result:

Signalmash will initially take a user to Volt page internal to its user portal, this page may show some basic description information about Volt Insights and the specific Signalmash / Volt offer, along with two buttons – SET UP VOLT and below it LOGIN TO VOLT. Before initial set-up is completed, the LOGIN TO VOLT option will be “grayed out” – when SET UP VOLT button is clicked, user will be taken to a second internal Signalmash hosted page where they will be shown the Terms & Conditions that they need to agree to, etc. and a radio button to agree along with a modal that says CONNECT ACCOUNT (or similar).

When CONNECT ACCOUNT button is clicked, a background call to Link URL Scheme will be made, user will be redirected to a page informing them

1) Your Signalmash account is being automatically connected with the Volt Insights platform. You will be able to login to Volt from the Signalmash platform once this process is completed.

2) You will receive an email from Volt to {Signalmash username email} set your password if you wish to login to the Volt Insights platform outside of the Signalmash portal.

Any return by user to the main Volt landing page after the CONNECT ACCOUNT action has occurred and before the Volt API call referenced in Step 2 below, should show the user that Volt Connection is pending and the LOGIN TO VOLT option should remain grayed out

Once all steps below are completed. User should see Connected to Volt Insights and the LOGIN TO VOLT option should now be “active” and no longer grayed.

When the Link URL Scheme is called after Volt -> Signalmash API call has occurred, user will be automatically logged in to Volt when the string is sent. This will be separate from the username / password login they would be able to perform directly to the Volt Insights portal.

Step 2: Volt sends a request to Signalmash identifying the user’s account as a Volt user.

Offline, Signalmash should send Volt documentation of an API endpoint and an API token.

Volt will send either a POST (for initial configuration) or a PATCH request (to update any of the volt specific paramaters) to the documented Signalmash URL, with a JSON body of this structure:

{

 "account\_id": 123,

 "user\_id": 123,

 "volt\_org\_id": 123,

 "volt\_user\_id": 123,

 "webhook\_url": "https://v1.insights.api.textvolt.com/data/messages/signalmash",

}

JSON body:

* account\_id — Signalmash account ID
* user\_id — Signalmash user ID
* volt\_org\_id — Volt organization ID
* volt\_user\_id — Volt user ID
* webhook\_url — The webhook URL endpoint for Signalmash webhooks for the unique account – there will not be any Authorization header requirement any longer

Step 3: Signalmash begins sending messaging-related webhooks to Volt.

Offline, Signalmash should send an example webhook body to Volt. Signalmash should send messaging-related webhooks to the URL provided for each account in step 2. The body of a webhook should include these values (values in italic aren’t strictly necessary but are preferred):

* *Message cost*
* *Message cost currency (unless all costs are USD)*
* Raw SMPP error code
* Raw SMPP delivery status
* *10DLC campaign ID (if available)*
* Unique message ID
* *ISO 8601 datetime the message was created within Signalmash (or, for inbound, the datetime when the message is received by Signalmash)*
* ISO 8601 datetime the message was sent downstream by Signalmash (or, for inbound, the datetime when the message is processed by Signalmash)
* ISO 8601 datetime the message was finalized (delivered/undelivered/timeout) (or, for inbound, the datetime when the message is processed by Signalmash)
* Origination phone number
* Termination phone number
* *Message body*
* Direction (inbound/outbound)
* Message type (SMS/MMS)
* *Segment count* (we can infer this if messages are GSM-7 encoded, but would prefer not to if this data is readily available)
* *Message size in bytes*
* *Customer price rate for the given message type (not including carrier fees)*
* *MCC of the termination phone number operator*
* *MNC of the termination phone number operator*
* *MNO name of the termination phone number operator*
* *AT&T campaign class*

Action Items

Signalmash should send to Volt:

* Public key from the key pair used to in URL signature
* API documentation for associating a Volt organization with a Signalmash account
* Signalmash API token (or equivalent)
* Example messaging-related webhook JSON body (or multiple examples, if the structure needs to differ in different contexts)

Signalmash should build:

* Link integrated into Signalmash web application
* API endpoint that Volt can use to associate a Signalmash account with a Volt organization
* A webhook sender for Volt-associated accounts

Volt should build:

* The URL to be embedded in Signalmash as a link and the associated functionality
	+ On initial use, sets up a Volt organization and notifies Signalmash API
	+ On subsequent use, automatically logs in
* Signalmash-specific webhook handler