**Verifyed**

**Data Model**

*Any credential Verifyed*

**Users**

* type: “Users” // required to group all Users documents into one association
* title
* firstname
* lastname
* password
* \*email (**Unique across shards)**
* \*registerIP
* \*status (Pending, Active, Suspended, Deleted, Review)
* \*source (site, FB/Twitter/Google/IN)
* referred\_by (for affiliates), default 0
* \*date\_created
* \*date\_updated
* timezoneoffset (time difference from GMT)
* \*updated\_by\_id (self or admin…)
* \*org\_id (0 if self or organization\_id from Organization collections)
* verification\_authorized\_count (past 7 days)
* Address

Street Address 1

Street Address 2

City/Region/Province

Country

Postal/Zip Code

* Contact

Home

Office

Cell/Mobile

FB/IN/Google/Tweet handle

* \*Login Details

Last login Time

Last Login ip

* \*Account Details

Balance

Last 3 Payments (date, amount, reference#)

Last 3 Charges (date, amount reference#)

db.runCommand( { shardCollection : "users" , key : { email : 1 } , unique : true } );

**Verifications**

* \_id (Mongo Object ID Auto generated)
* \*verify\_id (8 chars alphanumeric) **shard** ***unique index***
* \*requestor\_id (Organization id from Organization collection)
* \*requestor\_public\_key
* \*provider (Organization id from Organization collection)
* \*provider\_key\_id
* approver (user whose credential is being verified, approval optional)
* \*authorized\_by\_level1 (User id from User collection)
* \*authorized\_by\_supervisor (User id from User collection)
* \*date\_created
* \*date\_updated
* updated\_by\_id (\_id from users collections)
* type (credential, employment, reference, personal data)
* attachment
* description
* \*max\_auth\_pending\_days (set by requestor)
* \*date\_request\_processed
* \*status (New, Pending-Authorization, Pending-Acceptance, Request-Expired, Processed, Deleted, Disputed, Rejected)
* date\_disputed
* dispute\_reason (Wrong Authorization, Not adequate information, Information not genuine)
* date\_resolved
* dispute\_resolution\_medium (email, phone, customer service)
* dispute\_resolution\_code (Requestor provided wrong data, mutual agreement…)
* dispute\_resolution\_notes (user or CS may enter)

db.runCommand( { shardCollection : "verifications" , key : { verify\_id : 1 } , unique : true } );

**Disputes**

* \_id (Mongo Object ID Auto generated)
* \*verification\_id (id from verification collection)
* date\_disputed (same as date\_created)
* dispute\_reason (Wrong Authorization, Not adequate information, Information not genuine)
* date\_resolved
* dispute\_resolution\_medium (email, phone, customer service, online)
* dispute\_resolution\_code (Requestor provided wrong data, mutual agreement…)
* dispute\_resolution\_notes (user or CS may enter)

**Transactions**

* \_id (Mongo Object ID Auto generated)
* \*user\_id (id from from User Collection)
* \*verify\_id (id from Verification Collection)
* \*txn\_details (any non-payment information)
* \*date\_txn\_submitted
* \*reference\_code (an alphanumeric code required for transaction processing)
* \*date\_txn\_processed
* \*txn\_amount
* \*txn\_status (submitted, processed, declined)
* \*txn\_type (charge, refund, credit, chargeback, payout)
* \*auth\_code
* decline\_reason\_code

**Pricing (May use nools, based on rules engine)**

* \_id (Mongo Object ID Auto generated)
* country\_code
* currency\_code
* rule\_name
* rule\_type (fixed, range)
* rule
* creation date
* active (Yes|No)

**Accounts**

* \_id (Mongo Object ID Auto generated)
* \*organization\_id (organization\_id from Organizations Collections)
* \*verify\_id (id from Verification Collection, Everything account entry must be related to this, can be 0 for admin related)
* reference\_code (reference\_code from Transactions Collection)
* charge (For requesting verification in currency)
* credit (For providing verification)
* payout (cash out if balance is positive)
* entry\_type (charge, credit, payout, refund)
* entry\_reason (verification requested, verification provided, admin refund…)
* notes (anything to be shown to user)
* creation\_date
* deleted (default:0, 1 if an entry is marked deleted by admin)

**Organizations**

* org\_id (Universal ID, denoting the organization)
* \*name (Full, Florida Atlantic University)
* short\_name (FAU)
* ipaddress
* pubkey (for sharing with others)
* pvtkey (for opening documents encrypted with public key)
* Address

Street Address 1

Street Address 2

City/Region/Province

Postal/Zip Code

Country

* Contact

Name

Office Landline

Cell/Mobile

Email

designation

FB/IN/Google/Tweet handle

* \*type (Education, Enterprise, Agency, Board, Embassy)
* sub\_type (Education – University, College …, Enterprise – IT, Bank…, Agency – RTA…)
* signatories { id:1,2,3.., title:Director,Registrar…, contact{office, mobile, email}, role: signatory, supervisor…, Dept.}
* status (Pending, Active, Blacklisted, Withdrew…)
* \*Recent Activity

Verification Requested count (past 7 days)

Verification Provided count (past 7 days)

Alerts pending count

Messages unread count

* timezoneoffset (time difference from GMT)
* date\_created (UTC Miliseconds)
* date\_updated
* account\_balance (should always be updated)
* payout\_balance (+ve balance that can be withdrawn)

db.runCommand( { shardCollection : "organizations" , key : { organization\_id : 1 } , unique : true } );

**Analytics**