Scytl

Innovating Democracy

Bulletin Board

From theory to practice

E-Voting PhD Days, 2013, November 14–15, Münchenwiler, Switzerland

Jordi Cucurull Juan
R+D Department

Outline

- Verifiability of voting systems
- Bulletin Board
- An implementation
- Verifications
- Improvements

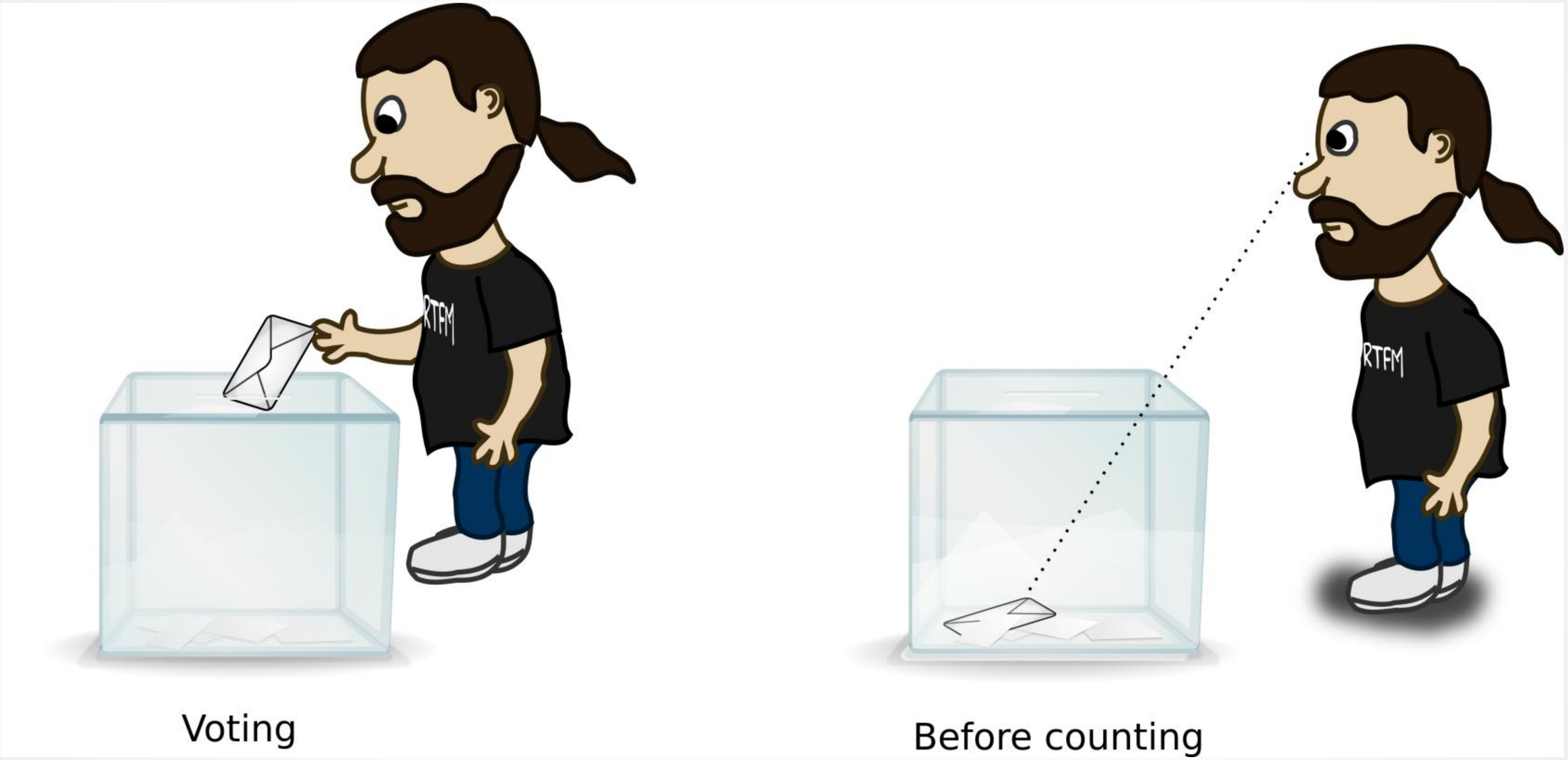
Verifiability of voting systems

Scytl
Innovating Democracy

End to End Verifiability

- Desirable verifiability properties
  - Cast as intended
  - Recorded as cast
  - Counted as recorded

Recorded as cast in voting paper



The voter can check his vote is present in the ballot box until the counting

Recorded as cast

How can the recorded as cast property be provided in electronic voting?

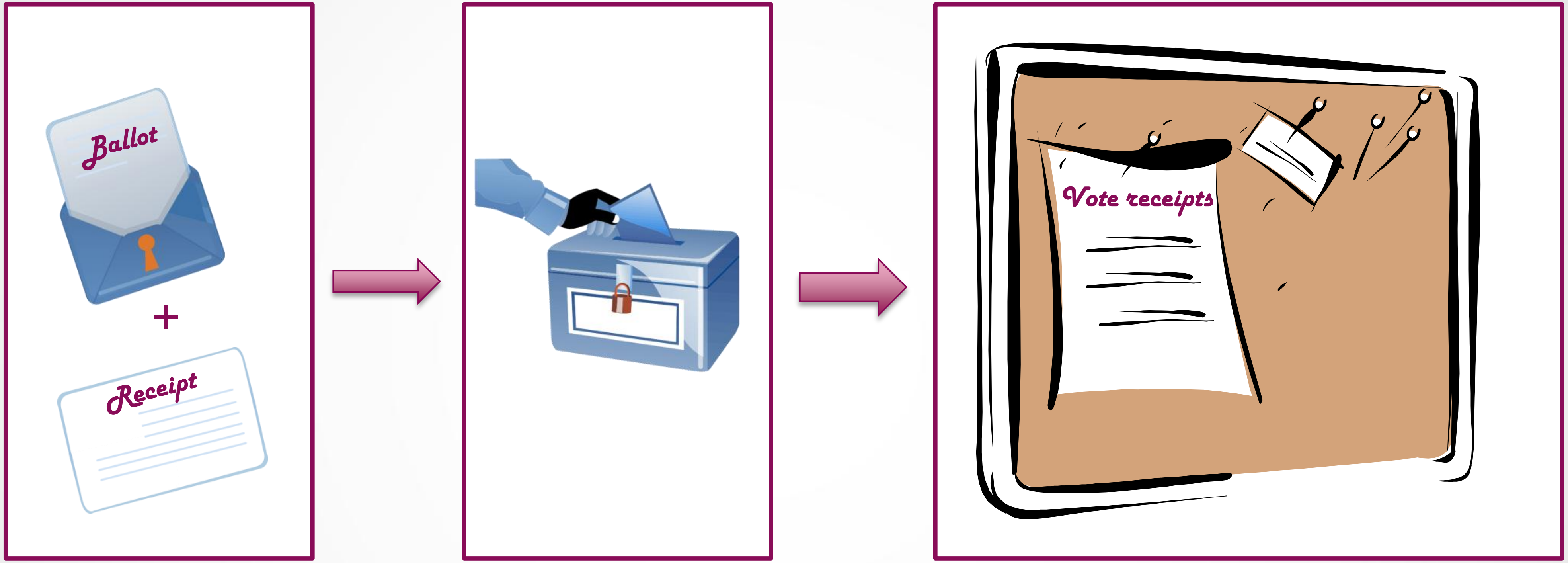
Scytl
Innovating Democracy

Bulletin Board

Bulletin Board

- Additional properties are required or desirable:
  - Public anonymous read-only access to the data
  - Append-only data writing by authorized parties

Recorded as cast in electronic voting

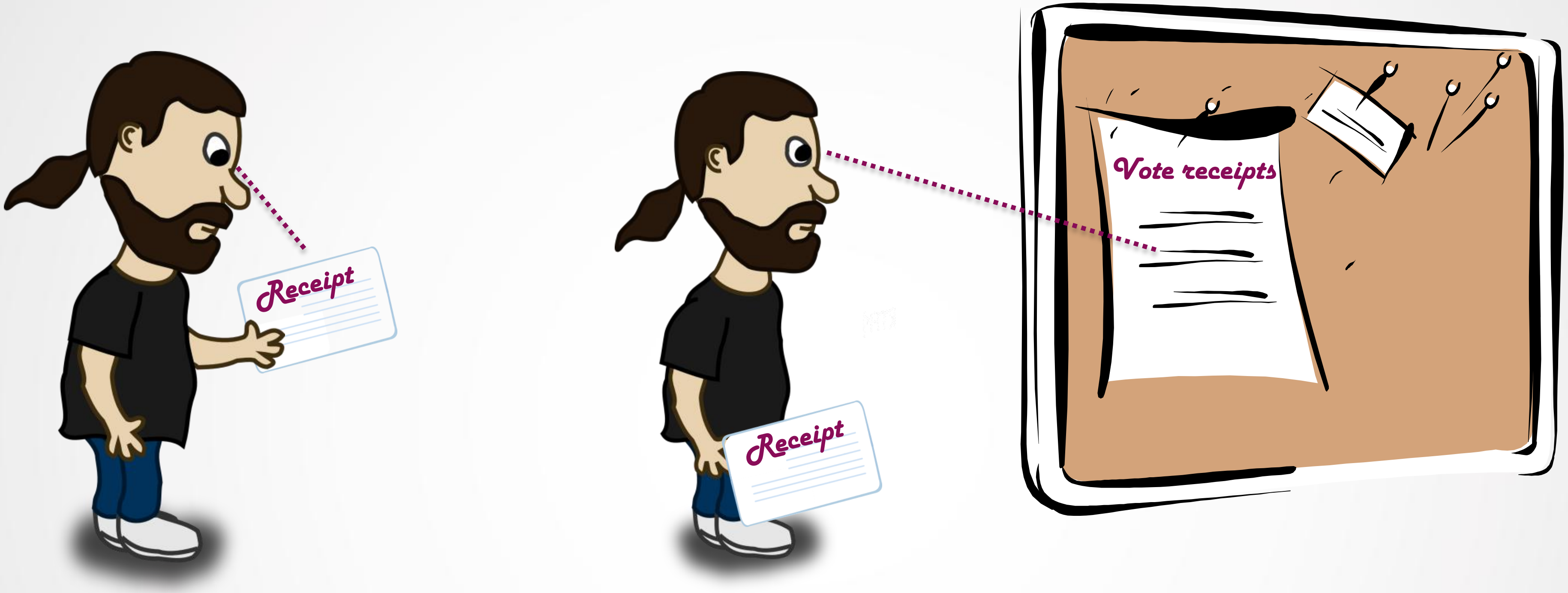


1.The voter generates a vote and a vote receipt (derived from the vote)

2.The voter casts the vote and keeps the receipt

3.The vote receipts derived from the votes in the Ballot Box are published in the Bulletin Board

Recorded as cast in electronic voting



4.The voter can check his voting receipt is present in the Bulletin Board

Design questions

- What data will be published?
- When will data be published?
- Who will publish in the Bulletin Board?
- Where will the Bulletin Board be located?
- How will data be presented?

What data will be published?

Vote receipt

Vote cast

- In both cases detached from voting options
(e.g. the vote must be randomly ciphered)

- Authenticity of data published guaranteed by cryptographic means
(i.e. signatures of the data)

When will data be published?

- Immediately
  - When a vote is cast the bulletin board is updated
- Periodically
  - The bulletin board is updated every a certain amount of time
- Quantitatively
  - The bulletin board is updated every a certain amount of votes cast

Who will publish in the Bulletin Board?

- Voting system component
  - Easy access to the ballot box
  - May require explicit ballot box access permission
  - Requires methods to deal with history changing
    - Prevention: append-only access to bulletin board
    - Detection: traceability of history changing

- External component
  - Independent of the voting system domain
  - Requires external read-only access to the ballot box
  - Risk of system intrusion and denial of service attack

Where will the Bulletin Board be located?

- Voting system domain
  - Voting system operators have full control of the data published
  - More susceptible to insider attacks, e.g. history modification

- Outside voting system domain
  - Voting system operators do not have full control of data published
  - Can be a commodity service or a specifically created service
  - Dependency with external systems

- Combination of both alternatives

How is data presented?

- Website
    - One or multiple lists with data
        - More difficult to access, easier to protect
    - Interactive form to search for data
        - Easier access, more difficult to protect

- Specific data service
    - FTP Server
    - Git Server
    - etc.

An implementation

eVote 2013

- Online voting system for Parliamentary 2013 elections in Norway

- Characteristics
  - Multi election system
  - Verifiable mixing and decryption
  - Provides cast as intended using return codes
  - Provides recorded as cast with a Bulletin Board
  - Implements Ballot Box data redundancy

Requirements and constraints

Requirements

- Bulletin Board for two functionalities
  - To support recorded as cast verifiability
  - To publish a proof of the contents of the Ballot Box

- Additional vote redundancy (two ballot boxes)

- Verification mechanism sound
  - Internal components must not be able to cheat without being detected

Constraints

- The voting client cannot be trusted

Design questions

- What will be published?
  - List of voting receipts as a snapshot of ballot box
  - Signatures of the receipts and the list
- When will data be published?
  - Periodically every a certain time period
- Who will publish in the Bulletin Board?
  - A voting system component
- Where will the Bulletin Board be located?
  - Combined approach with an internal and an external Git repository
- How will data be presented?
  - Git Hub service
    - Public access to last revision through website
    - Public access to the whole repository, i.e. revisions of the list

Scytl
Innovating Democracy

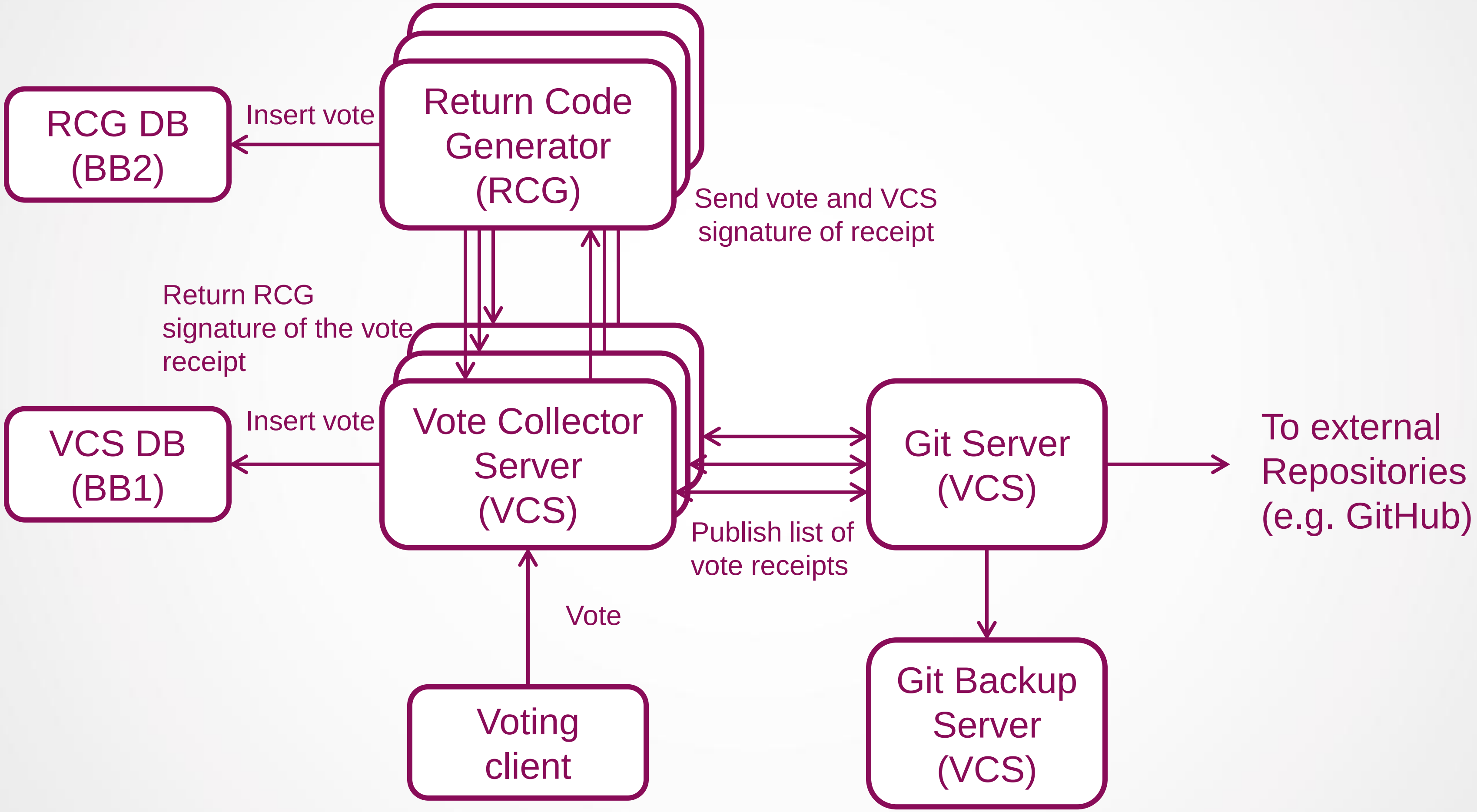
Why a Git repository?

- List of receipts periodically published and stored

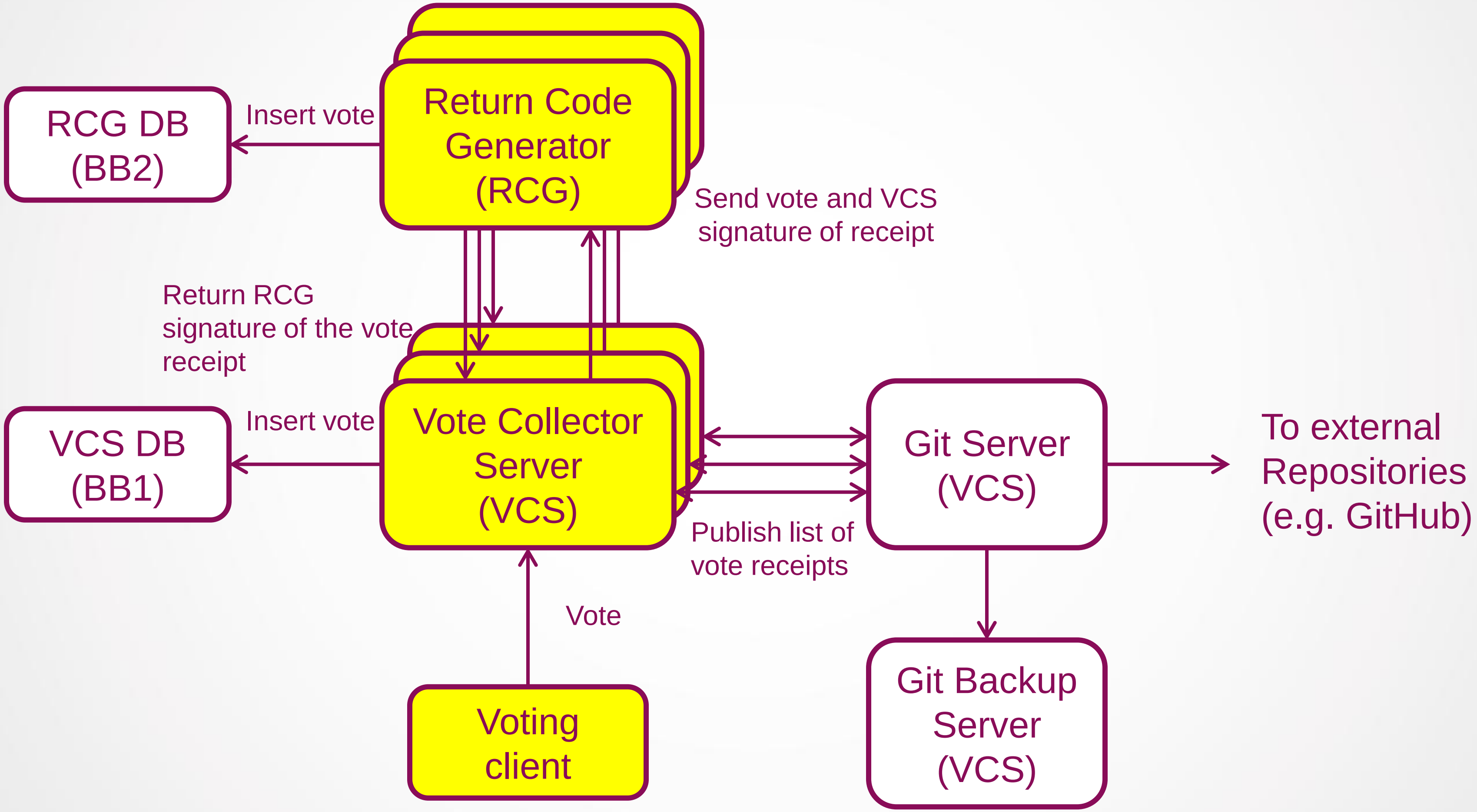
- Git provides
  - Off-the-shelf distributed revision control of files
    - Repository easy to replicate
    - History of the data modifications
  - Compatibility with third party repository services, e.g. GitHub

- Cons
  - History is not cryptographically secure
    (auditors have to keep track of revisions)
  - Deltas are not supported (compression instead)
  - Constrained repository and file size
    - GitHub max repository: 1GB with revisions
    - GitHub max file size: 50-100 MB (enough for 250.000 votes)

eVote 2013 – Voting components


RCG DB
(BB2)
Insert vote
Return Code Generator (RCG)
Send vote and VCS signature of receipt
Return RCG signature of the vote receipt
VCS DB
(BB1)
Insert vote
Vote Collector Server (VCS)
Git Server (VCS)
To external Repositories (e.g. GitHub)
Publish list of vote receipts
Vote
Voting client
Git Backup Server (VCS)

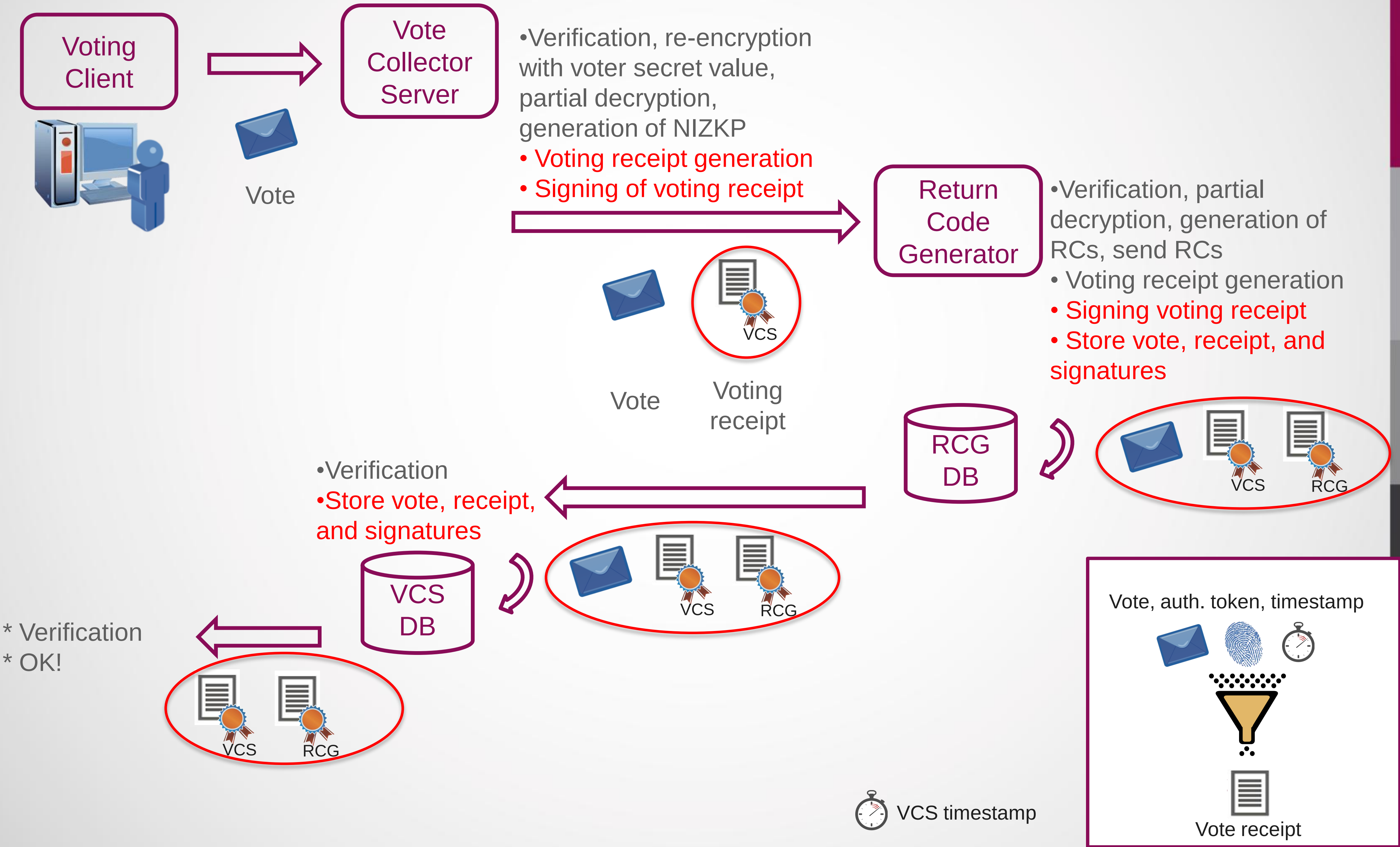
Electronic voting protocol



Return Code Generator (RCG)
RCG DB (BB2)
Insert vote
Send vote and VCS signature of receipt
Return RCG signature of the vote receipt
VCS DB (BB1)
Insert vote
Vote Collector Server (VCS)
Git Server (VCS)
To external Repositories (e.g. GitHub)
Publish list of vote receipts
Vote
Voting client
Git Backup Server (VCS)

Scytl
Innovating Democracy

Electronic voting protocol



Voting Client

Vote

Vote Collector Server

- Verification, re-encryption with voter secret value, partial decryption, generation of NIZKP
- Voting receipt generation
- Signing of voting receipt

Vote

VCS

Voting receipt

Return Code Generator

- Verification, partial decryption, generation of RCs, send RCs
- Voting receipt generation
- Signing voting receipt
- Store vote, receipt, and signatures

RCG DB

VCS

RCG

- Verification
- Store vote, receipt, and signatures

VCS DB

VCS

RCG

* Verification
* OK!

VCS

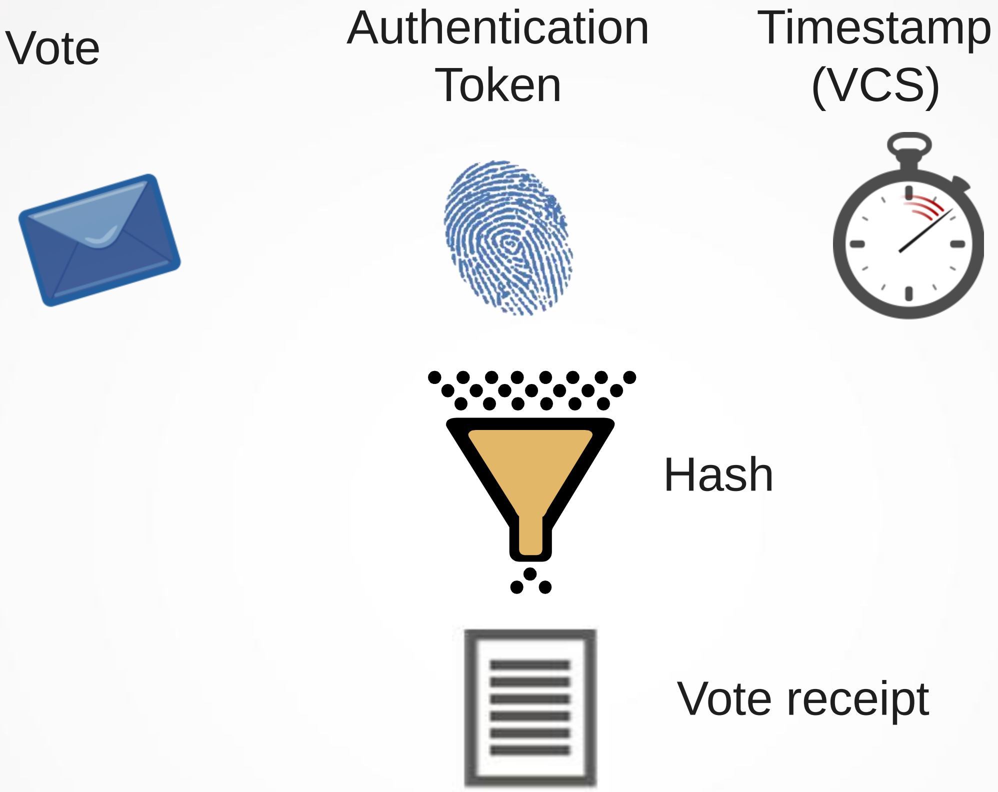
RCG

Vote, auth. token, timestamp

Vote receipt

VCS timestamp

Voting receipt



Vote

Authentication Token

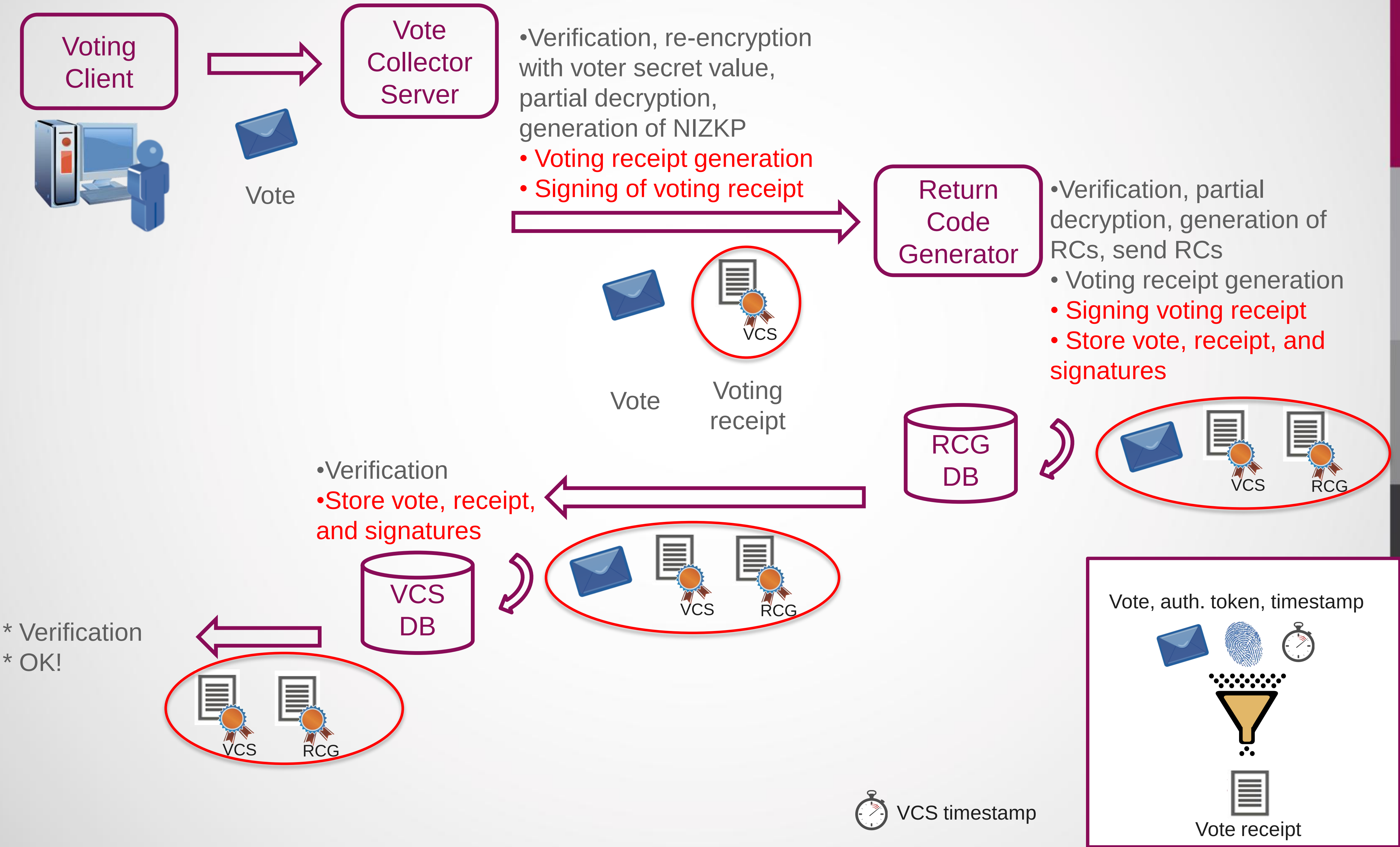
Timestamp (VCS)

Hash

Vote receipt

- Vote receipt is unique
    - Vote is randomly ciphered
    - The authentication token can only be used once for a given election
    - The timestamp is created at the reception of the vote

Electronic voting protocol



•Verification, re-encryption with voter secret value, partial decryption, generation of NIZKP

• Voting receipt generation

• Signing of voting receipt

•Verification, partial decryption, generation of RCs, send RCs

• Voting receipt generation

• Signing voting receipt

• Store vote, receipt, and signatures

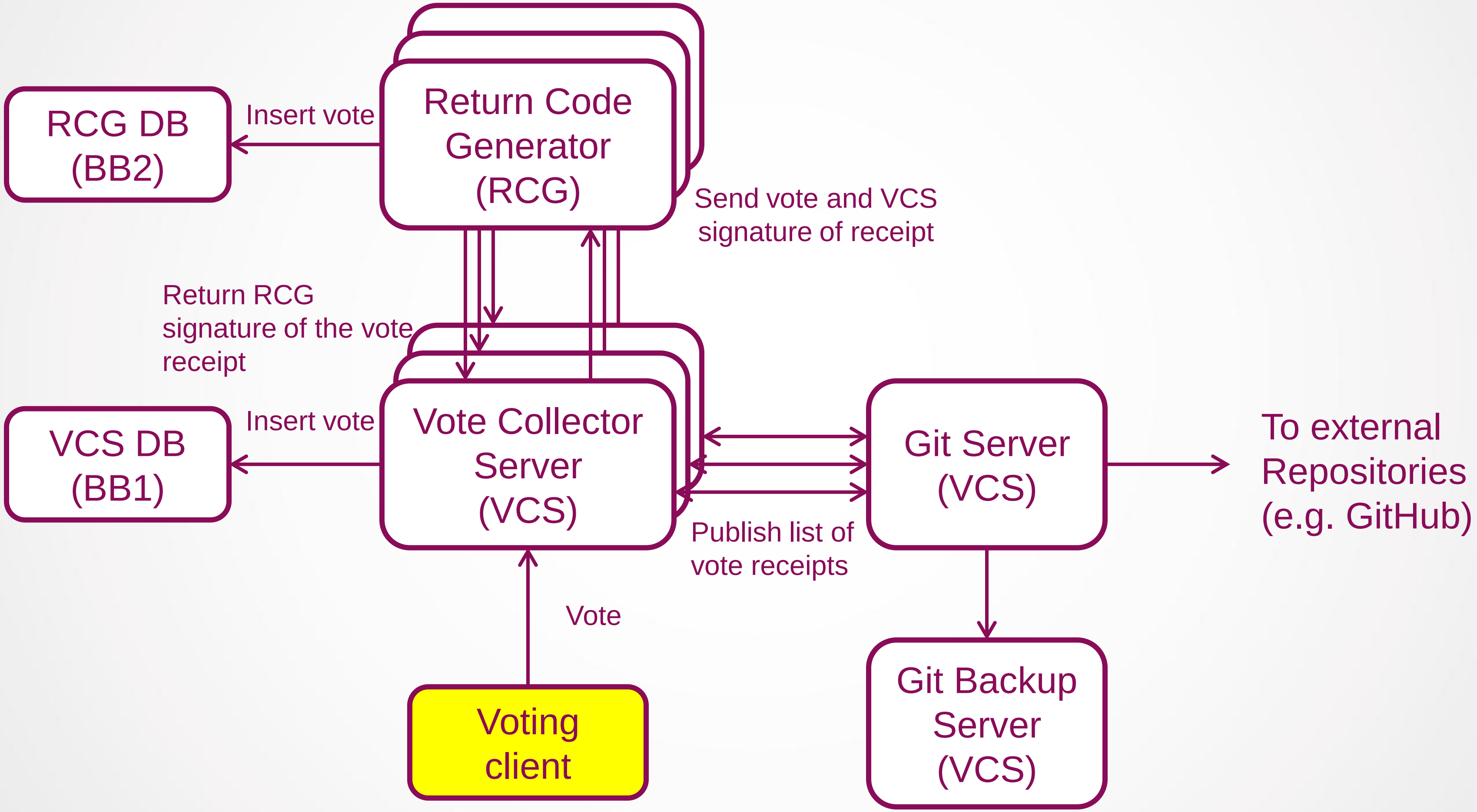
•Verification

•Store vote, receipt, and signatures

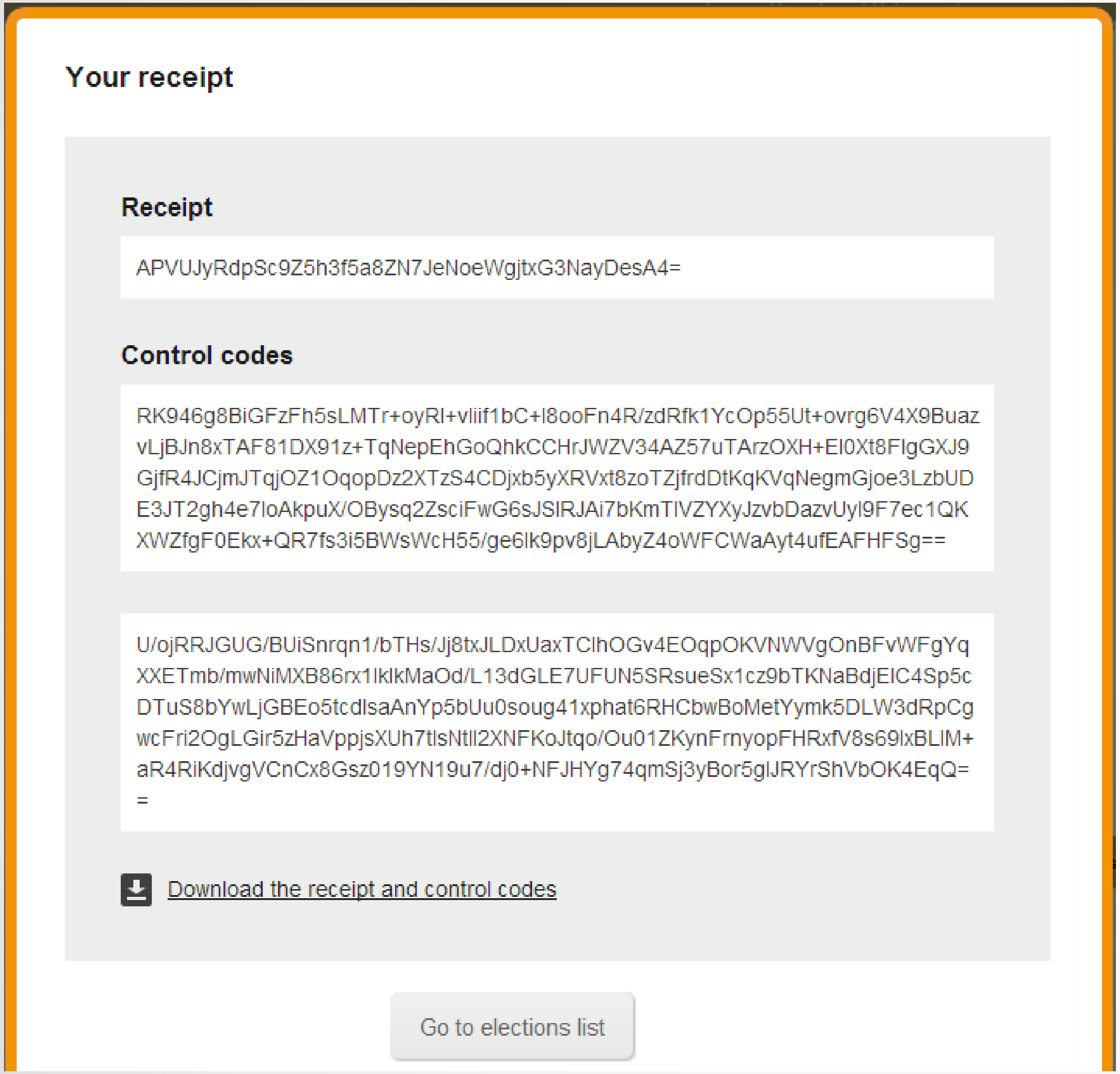
* Verification

* OK!

Voting client modifications



Vote receipt given to the voter



Your receipt

Receipt

APVUJyRdpSc9Z5h3f5a8ZN7JeNoeWgjtxG3NayDesA4=

Control codes

RK946g8BiGFzFh5sLMTr+oyRl+vliif1bC+l8ooFn4R/zdRfk1YcOp55Ut+ovrg6V4X9Buaz
vLjBJn8xTAF81DX91z+TqNepEhGoQhkCCHrJWZV34AZ57uTArzOXH+El0Xt8FlgGXJ9
GjfR4JCjmJTqjOZ1OqopDz2XTzS4CDjxb5yXRVxt8zoTZjfrdDtKqKVqNegmGjoe3LzbUD
E3JT2gh4e7loAkpuX/OBysq2ZsciFwG6sJSlRJAi7bKmTlVZYXyJzvbDazvUyl9F7ec1QK
XWZfgF0Ekx+QR7fs3i5BWsWcH55/ge6lk9pv8jLAbyZ4oWFCWaAyt4ufEAFHFSg==

U/ojRRJGUG/BUiSnrqn1/bTHs/Jj8txJLDxUaxTClhOGv4EOqpOKVNWVgOnBFvWFgYq
XXETmb/mwNiMXB86rx1lklkMaOd/L13dGLE7UFUN5SRsueSx1cz9bTKNaBdjElC4Sp5c
DTuS8bYwLjGBEo5tcdlsaAnYp5bUu0soug41xphat6RHCbwBoMetYymk5DLW3dRpCg
wcFri2OgLGir5zHaVppjsXUh7tlsNtll2XNFKoJtqo/Ou01ZKynFrnyopFHRxfV8s69lxBLlM+
aR4RiKdjvgVCnCx8Gsz019YN19u7/dj0+NFJHYg74qmSj3yBor5glJRYrShVbOK4EqQ=
=

⬇ Download the receipt and control codes

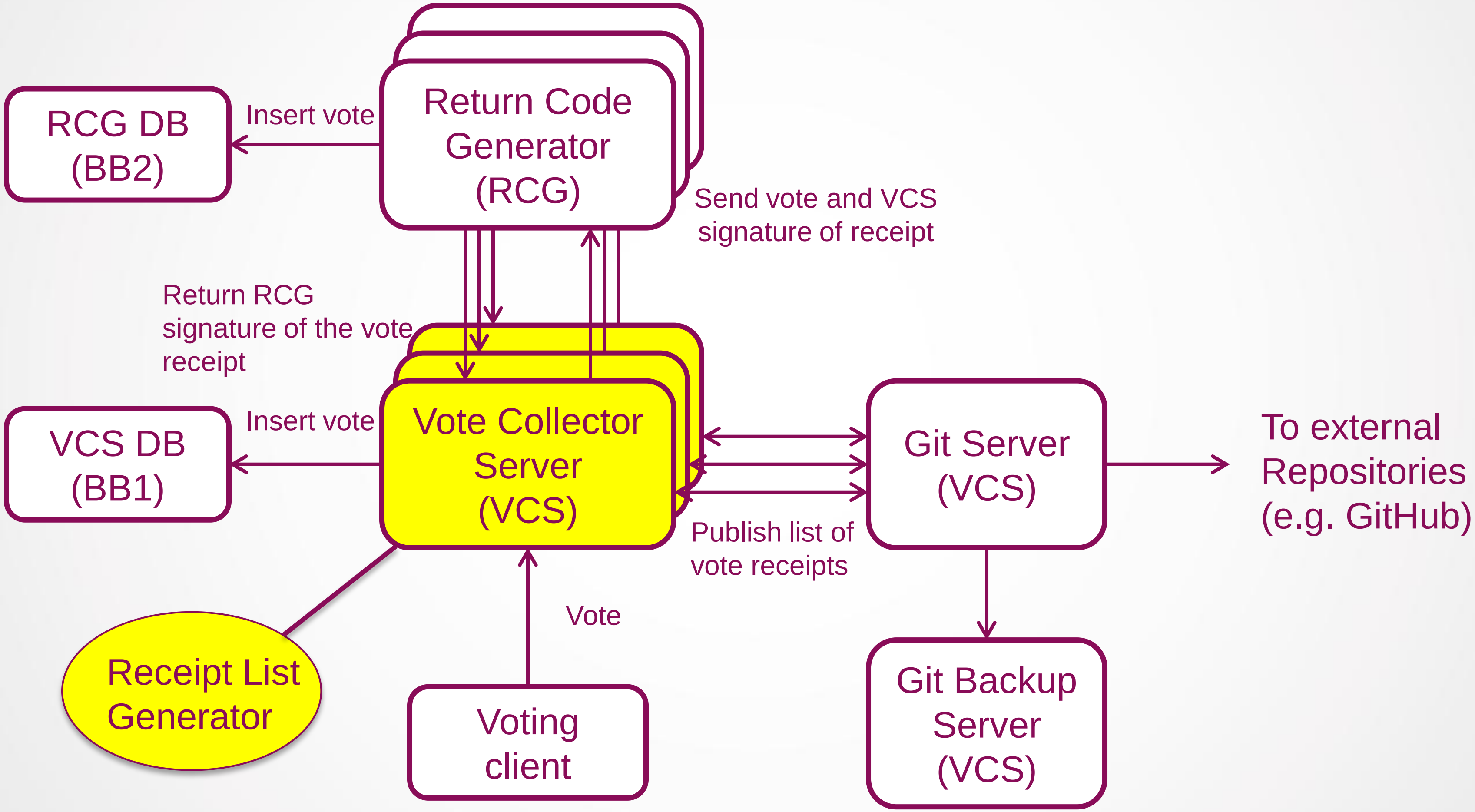
Hash of vote

RCG signature of receipt

VCS signature of receipt

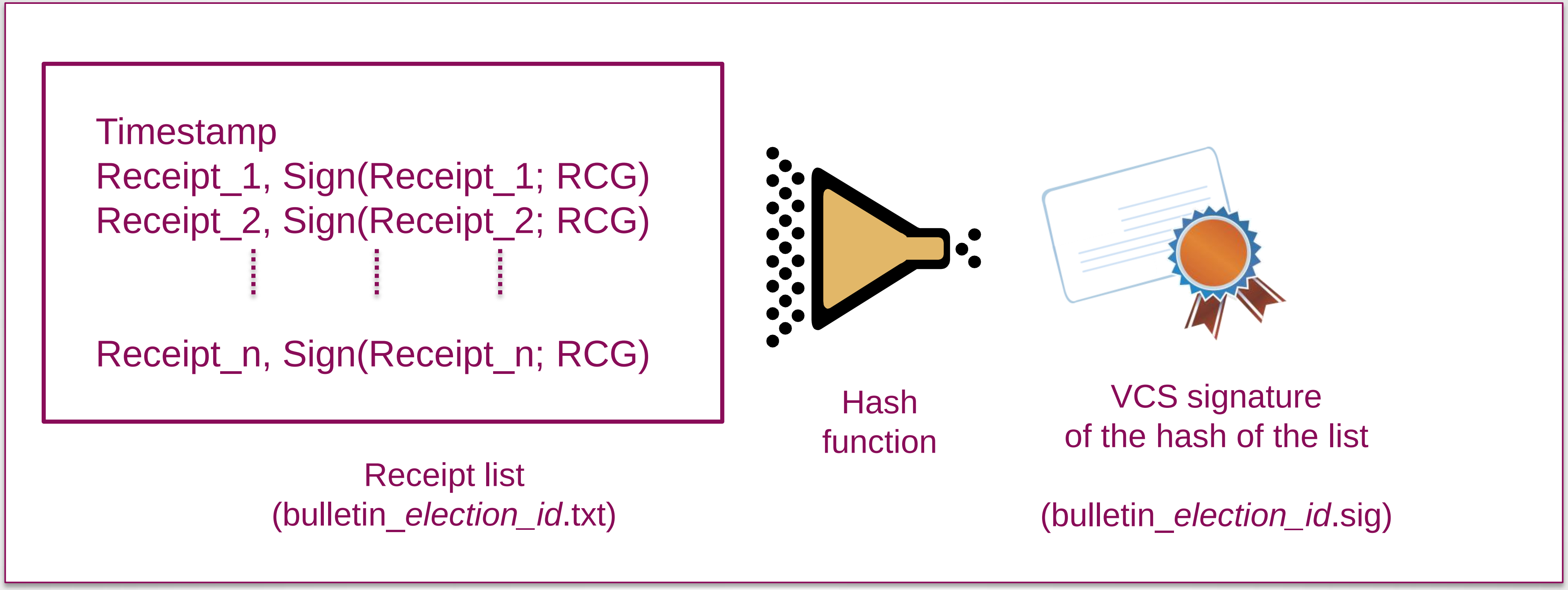
Signatures guarantee
the vote receipt is authentic

Addition of the Receipt List Generator



RCG DB (BB2)
Insert vote
Return Code Generator (RCG)
Send vote and VCS signature of receipt
Return RCG signature of the vote receipt
VCS DB (BB1)
Insert vote
Vote Collector Server (VCS)
Git Server (VCS)
To external Repositories (e.g. GitHub)
Publish list of vote receipts
Vote
Receipt List Generator
Voting client
Git Backup Server (VCS)

Receipt list and signature



Timestamp
Receipt_1, Sign(Receipt_1; RCG)
Receipt_2, Sign(Receipt_2; RCG)
⋮ ⋮ ⋮
Receipt_n, Sign(Receipt_n; RCG)

Receipt list
(bulletin_*election_id*.txt)

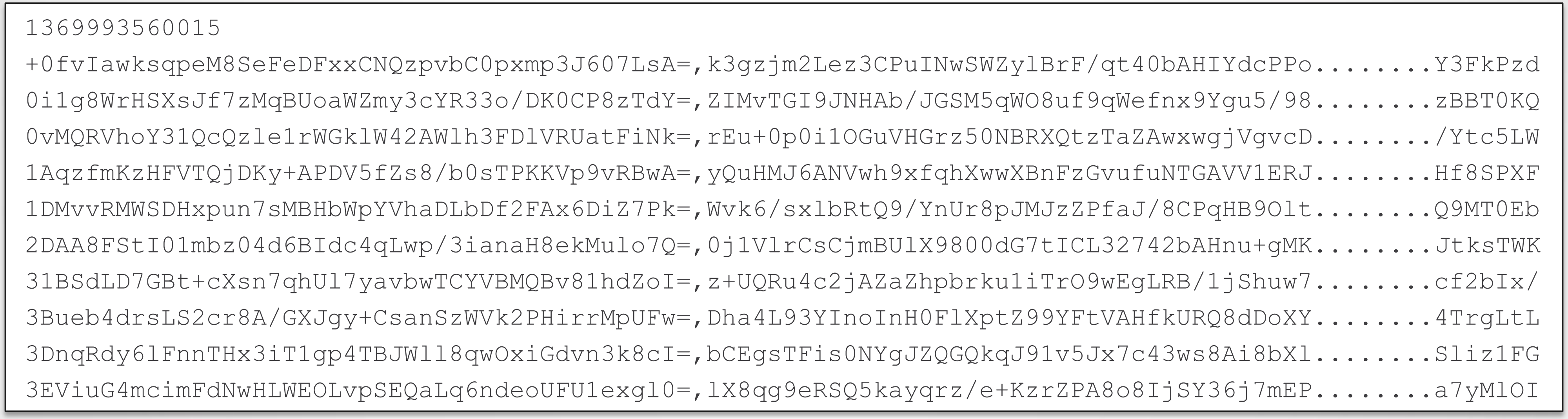
Hash
function

VCS signature
of the hash of the list

(bulletin_*election_id*.sig)

- Non incremental list of receipts periodically generated
- RCG individual receipt signatures state the votes were cast in RCG DB
- VCS signature states the votes are in VCS DB

Example of receipt list

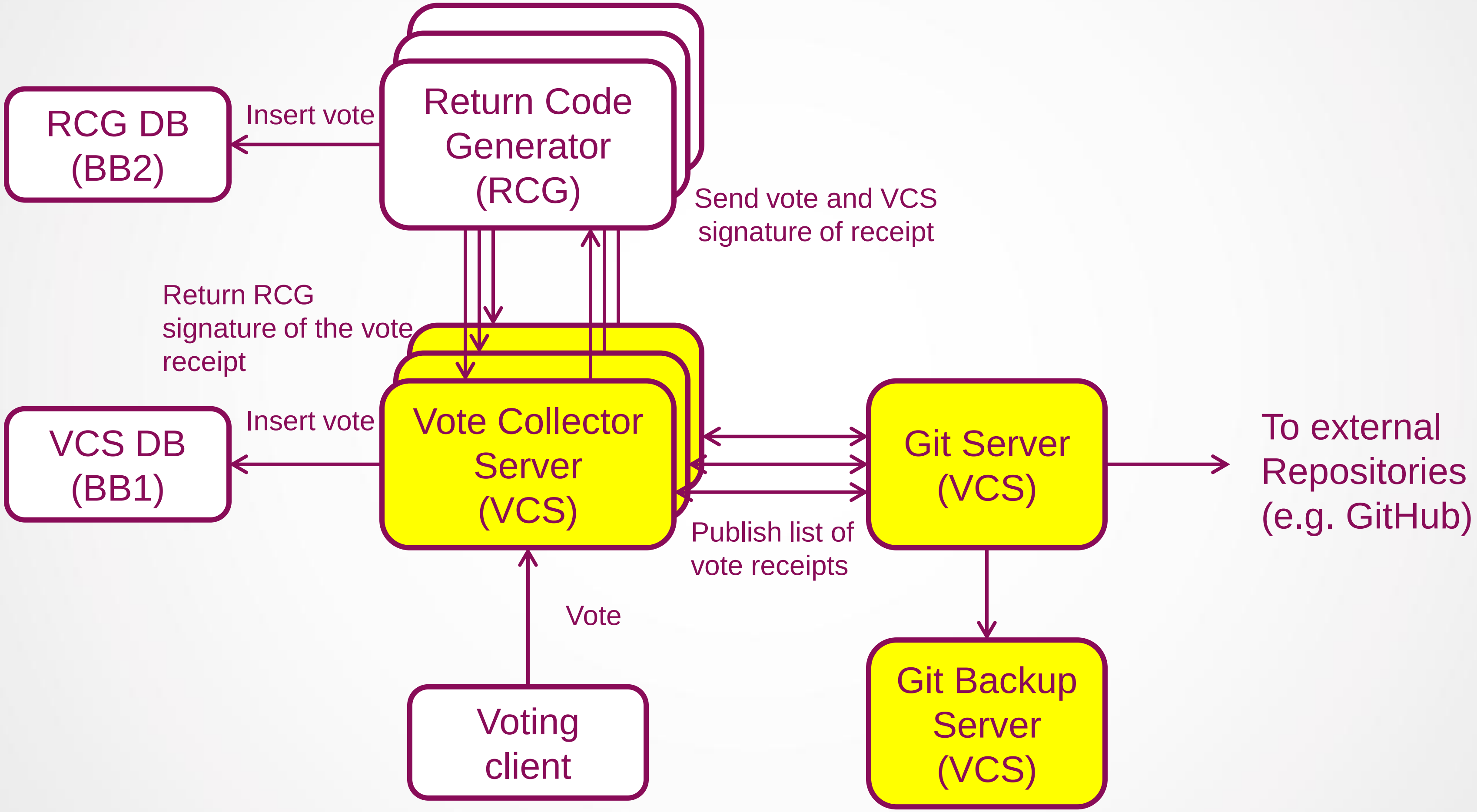


```
1369993560015
+0fvIawksqpeM8SeFeDFxxCNQzpvbC0pxmp3J607LsA=,k3gzjm2Lez3CPuINwSWZylBrF/qt40bAHIYdcPPo........Y3FkPzd
0i1g8WrHSXsJf7zMqBUoaWZmy3cYR33o/DK0CP8zTdY=,ZIMvTGI9JNHAb/JGSM5qWO8uf9qWefnx9Ygu5/98........zBBT0KQ
0vMQRVhoY31QcQzle1rWGklW42AWlh3FDlVRUatFiNk=,rEu+0p0i1OGuVHGrz50NBRXQtzTaZAwxwgjVgvcD......../Ytc5LW
1AqzfmKzHFVTQjDKy+APDV5fZs8/b0sTPKKVp9vRBwA=,yQuHMJ6ANVwh9xfqhXwwXBnFzGvufuNTGAVV1ERJ........Hf8SPXF
1DMvvRMWSDHxpun7sMBHbWpYVhaDLbDf2FAx6DiZ7Pk=,Wvk6/sxlbRtQ9/YnUr8pJMJzZPfaJ/8CPqHB9Olt........Q9MT0Eb
2DAA8FStI01mbz04d6BIdc4qLwp/3ianaH8ekMulo7Q=,0j1VlrCsCjmBUlX9800dG7tICL32742bAHnu+gMK........JtksTWK
31BSdLD7GBt+cXsn7qhUl7yavbwTCYVBMQBv81hdZoI=,z+UQRu4c2jAZaZhpbrku1iTrO9wEgLRB/1jShuw7........cf2bIx/
3Bueb4drsLS2cr8A/GXJgy+CsanSzWVk2PHirrMpUFw=,Dha4L93YInoInH0FlXptZ99YFtVAHfkURQ8dDoXY........4TrgLtL
3DnqRdy6lFnnTHx3iT1gp4TBJWll8qwOxiGdvn3k8cI=,bCEgsTFis0NYgJZQGQkqJ91v5Jx7c43ws8Ai8bXl........Sliz1FG
3EViuG4mcimFdNwHLWEOLvpSEQaLq6ndeoUFU1exgl0=,lX8qg9eRSQ5kayqrz/e+KzrZPA8o8IjSY36j7mEP........a7yMlOI
```

- The list is sorted by alphabetical order of the receipt
    - To avoid presenting the votes by time correlation

- The voter has to search her receipt in the list

Addition of a Git Server



RCG DB (BB2)

Insert vote

Return Code Generator (RCG)

Send vote and VCS signature of receipt

Return RCG signature of the vote receipt

VCS DB (BB1)

Insert vote

Vote Collector Server (VCS)

Git Server (VCS)

To external Repositories (e.g. GitHub)

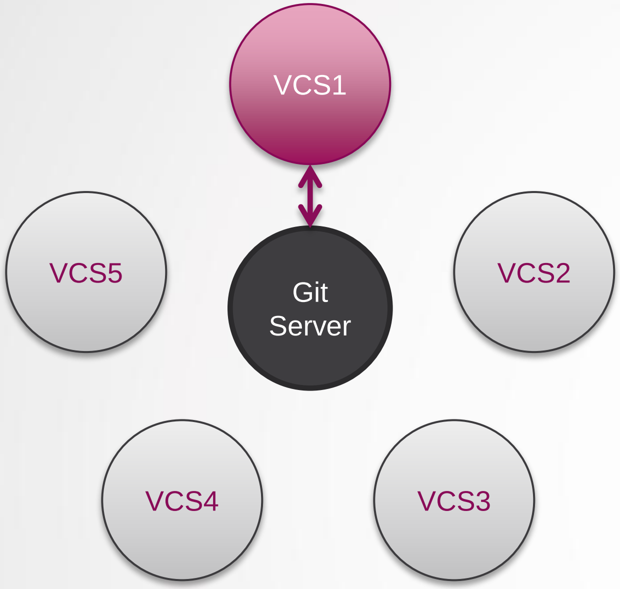
Publish list of vote receipts

Vote

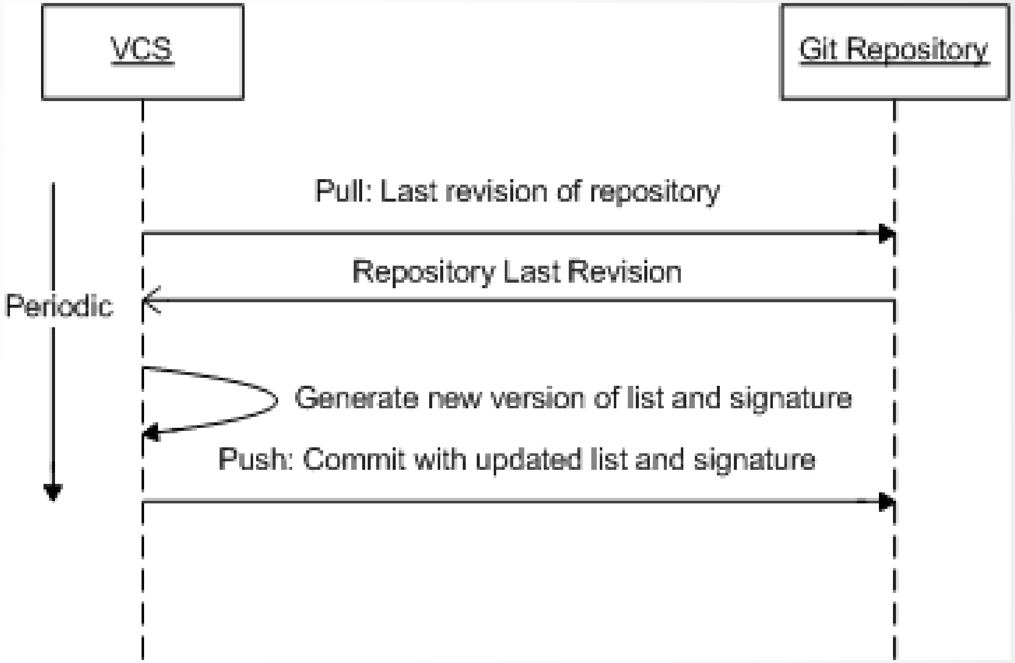
Voting client

Git Backup Server (VCS)

Publication of the list using a Git repository



Infrastructure


VCS
Git Repository
Pull: Last revision of repository
Repository Last Revision
Periodic
Generate new version of list and signature
Push: Commit with updated list and signature

Publication of the list

- Fault tolerant mechanism to generate and publish the list
  - Only one node is selected at a time to perform the task

- Requires synchronisation with a Git repository
  - Currently this is the Git Server inside one of the VCSs

Retrieval of the list and/or repository

- Retrieval of the list (GitHub service or similar)
    - The last revision of the list can be obtained via HTTP
    - The last revision of the list and signature can be retrieved into a ZIP file
    - Example of address:

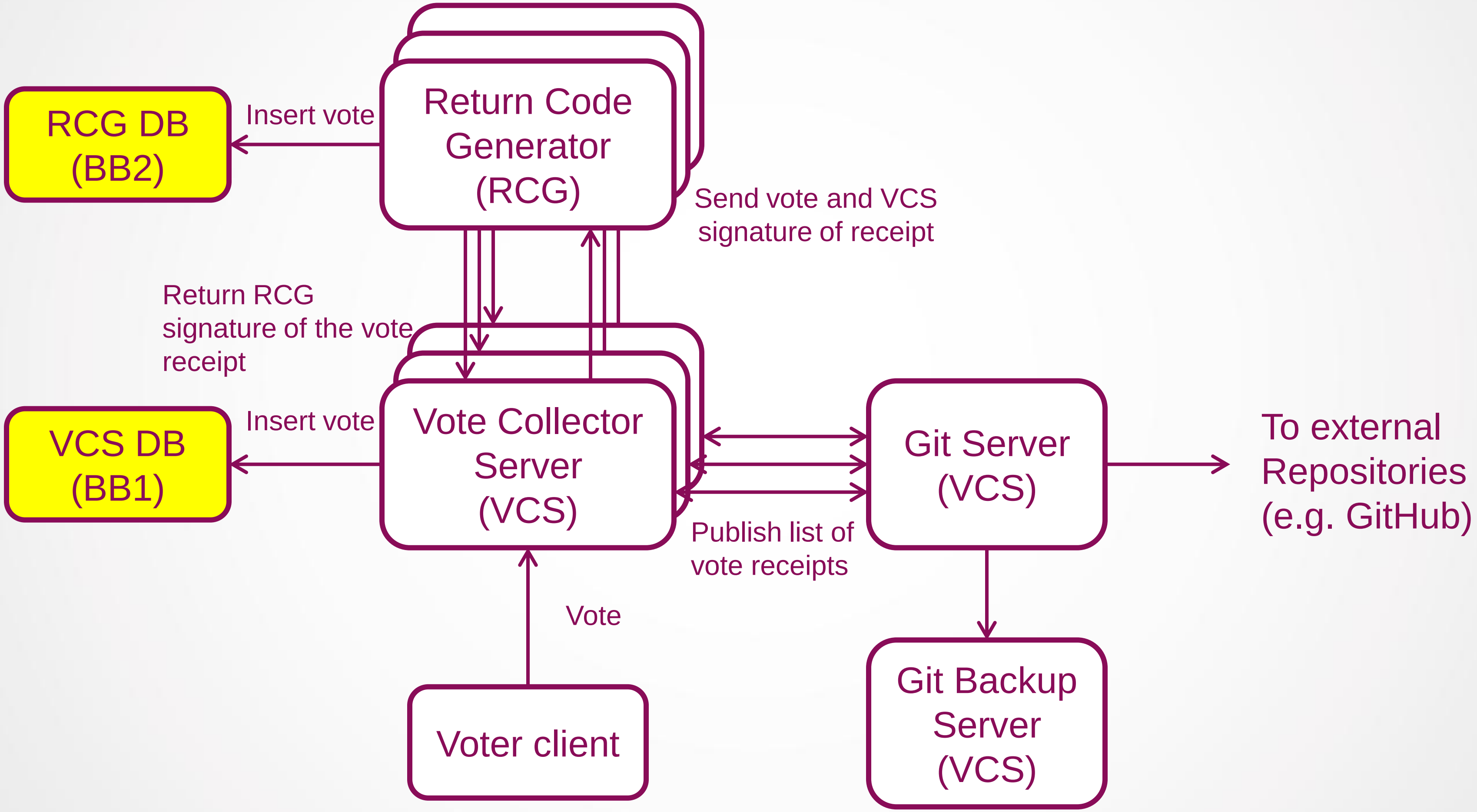
        https://github.com/user/repository/archive/master.zip

- Retrieval of the repository
    - The whole repository can be cloned
    - This includes all the list revisions (non cryptographically secure)
    - Example of address:

        git://github.com/user/repository.git

Scytl
Innovating Democracy

Ballot Box redundancy


RCG DB (BB2)
Insert vote
Return Code Generator (RCG)
Send vote and VCS signature of receipt
Return RCG signature of the vote receipt
VCS DB (BB1)
Insert vote
Vote Collector Server (VCS)
Git Server (VCS)
To external Repositories (e.g. GitHub)
Publish list of vote receipts
Vote
Voter client
Git Backup Server (VCS)

Ballot Box redundancy

- Votes can be lost from a ballot box for several reasons
  - Technical failures
  - Human errors during management
  - Attacks to the ballot box

- Bulletin Board allows the detection of lost votes

- Ballot Box redundancy allows the recovering of lost votes

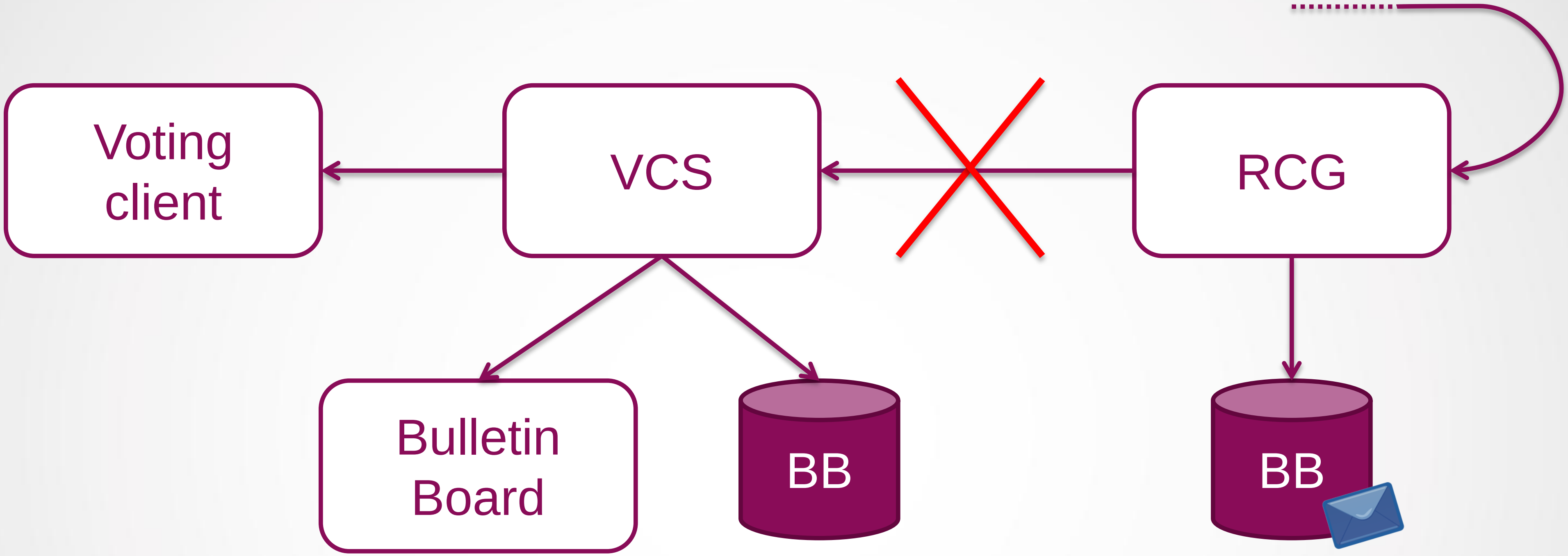
Ballot Box data coherency

- Ballot Box redundancy introduces data coherency issues

- What if a vote is only in one of the ballot boxes?
  - In a perfect world the vote should be counted
  - In our system in some cases the vote must not be counted

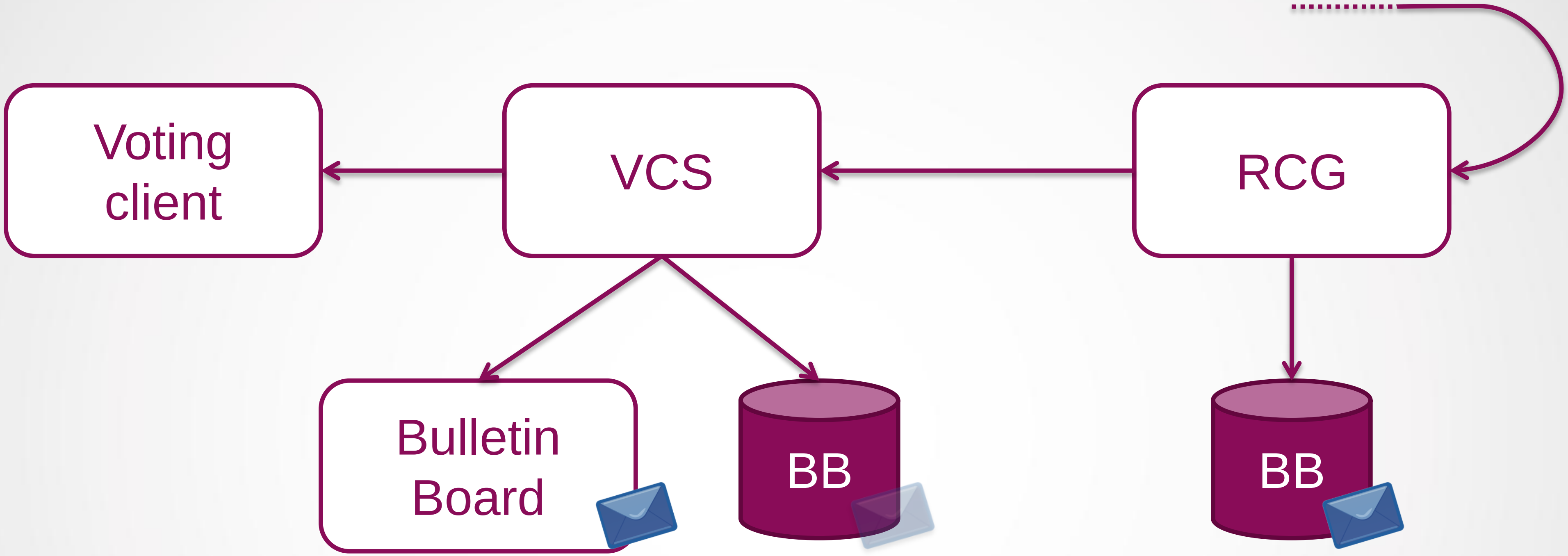
- But the voting system is distributed and subject to network errors!!!
  - The vote is not atomically cast in both ballot boxes
  - A vote is considered cast when the user is notified about it
    - This only happens after casting the vote in VCS DB
    - This happens via voting client

Vote lost due to communication issue



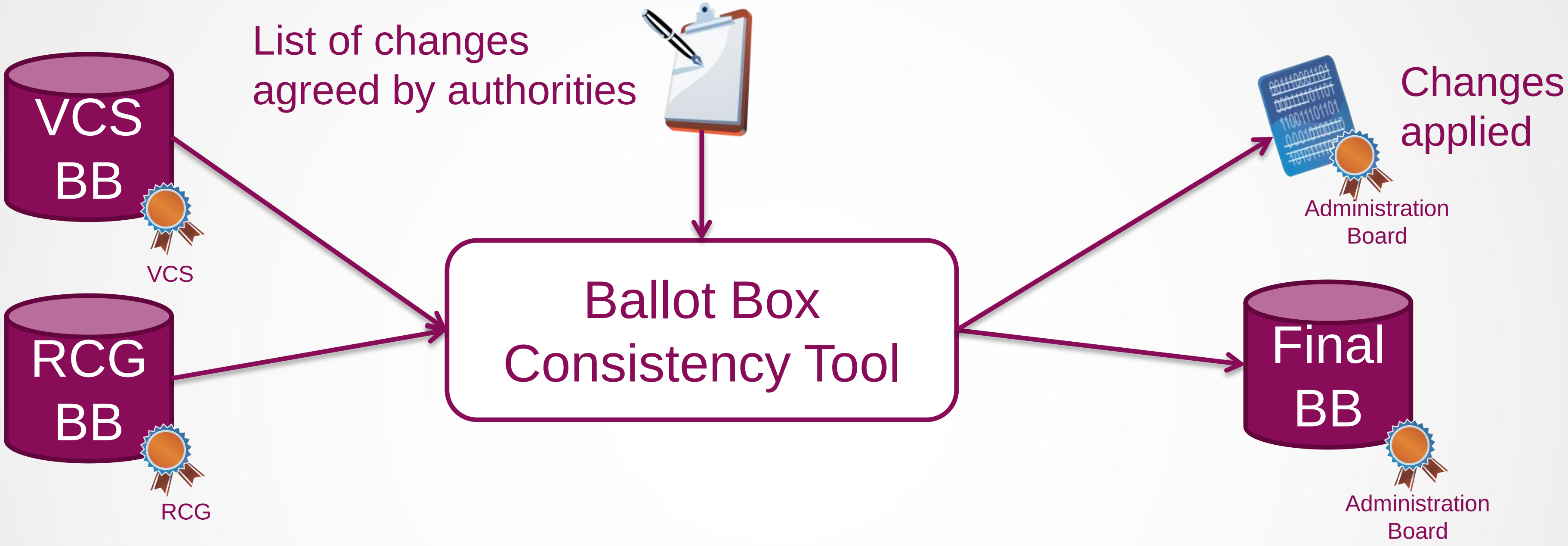
- Communication failure between RCG and VCS
  - In this case the vote was never confirmed to the user
  - Hence the vote **must not be counted**

Vote lost due to attack or failure in the ballot box



- Attack or technical problem
  - If vote appears in Bulletin Board means it was confirmed to the user
  - The **vote must be counted**

Adjustment of coherency



- Ballot Box Consistency Tool
  - Generates a new signed Ballot Box for the cleansing
  - Generates a signed report of the changes for traceability
  - Everything is signed by the Administration Board

Verifications

Verification of the published list

List authenticity validation

1. Extract public key of VCS from the certificate

```
$ openssl x509 -pubkey -noout -in certificateVCS.crt > pubkeyVCS.pem
```

2. Perform the SHA256 of the receipts list

```
$ sha256sum bulletin_election_id.txt | grep -o '^[^ ]*' | tr -d '\n' > sha.txt
```

3. Validate the signature of the list with the SHA256 obtained using the VCS public key

```
$ openssl dgst -sha256 -verify pubkeyVCS.pem -signature
     bulletin_election_id.sig sha.txt
```

Verification of the published list

Receipt authenticity validation

1. Extract the public key of the RCG certificate:

```
$ openssl x509 -pubkey -noout -in certificateRCG.crt > pubkeyRCG.pem
```

2. Get next entry ($entry) of the list

3. Extract receipt of the entry:

```
$ echo $entry | awk -F',' '{print $1}' | grep -o '^[^ ]*' | tr -d '\n'  > receipt.txt
```

4. Extract the signature of the receipt (converting it from Base64 to binary):

```
$ echo $entry | awk -F',' '{print $2}' | base64 -d > receipt.sig
```

5. Validate the signature of the receipt:

```
$ openssl dgst -sha256 -verify pubkeyRCG.pem -signature receipt.sig receipt.txt
```

6. Goto step 2 to validate the next entry.

Scytl
Innovating Democracy

Improvements

Improvements

- Repository separated from VCS (dedicated server)
- Chaining of produced lists of receipts
- Replacement of the Git protocol for supporting deltas
- Front end service for easing usability
- Automatic crosscheck of votes and receipts in ballot box before publication
- Bulletin Board supporting different elections in different repositories

Conclusions

Conclusions

- A Bulletin Board can be used to provide recorded as cast verifiability

- The design of a Bulletin Board is not a simple task
    - A large number of design decisions have to be taken

- A complete Bulletin Board has been implemented for eVote 2013
    - Design of vote receipt
    - Delivery of voting receipt to the voter
    - Generator of receipt lists
    - Infrastructure to maintain and publish the lists
    - Ballot box redundancy

- A number of improvements are proposed to following releases

Innovating Democracy