# Internet voting by CHvote

## Lessons learnt and future work

REPUBLIQUE
ET CANTON
DE GENEVE

POST TENEBRAS LUX

**Département de la sécurité et de l'économie**
**Direction générale des systèmes d'information**

# Short Bio

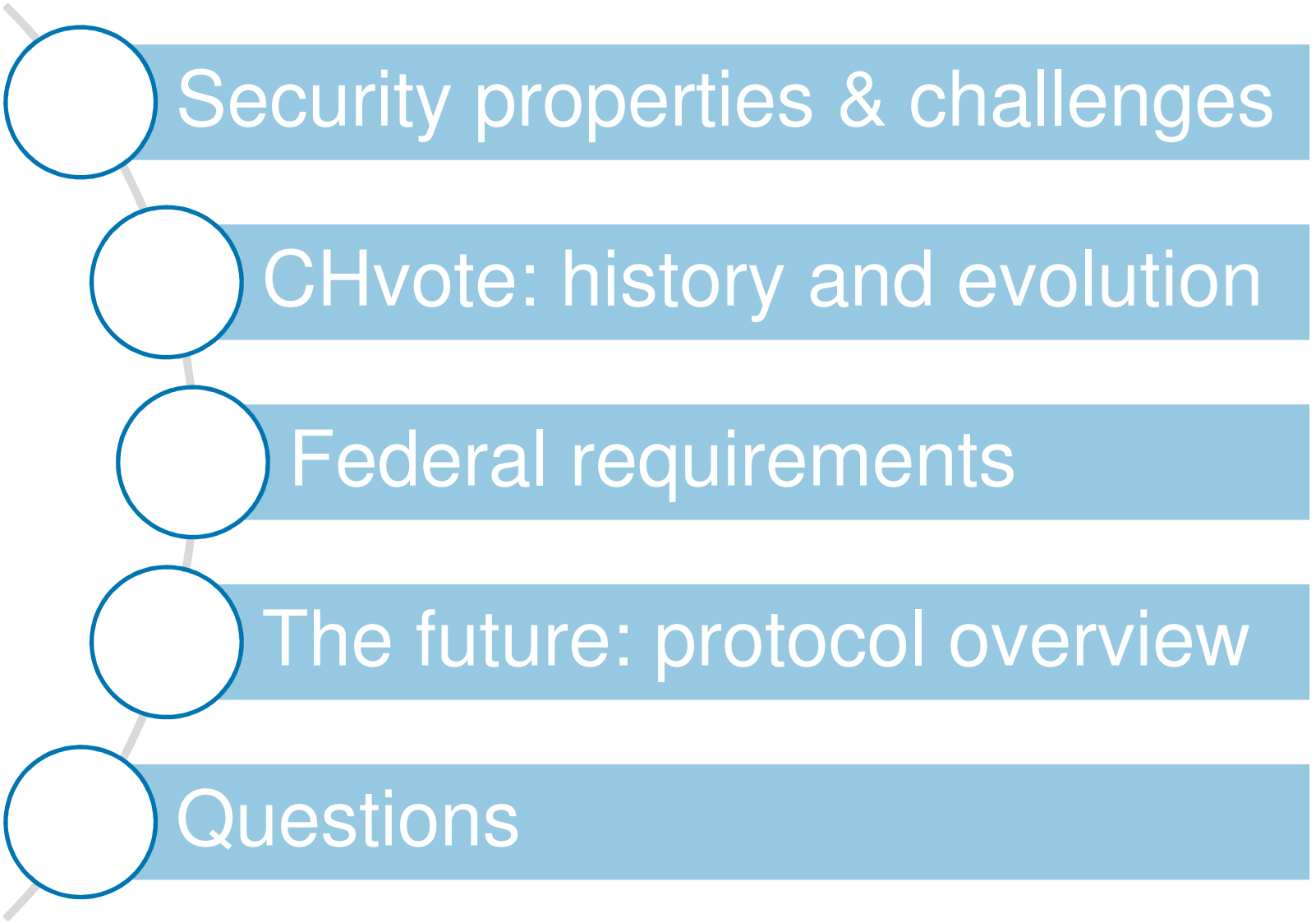
Thomas Hofer / @thhofer / thomas.hofer@etat.ge.ch

- EPFL MSc in IT
- Master thesis @ CERN Computer Security Team
- IT / Java consultant

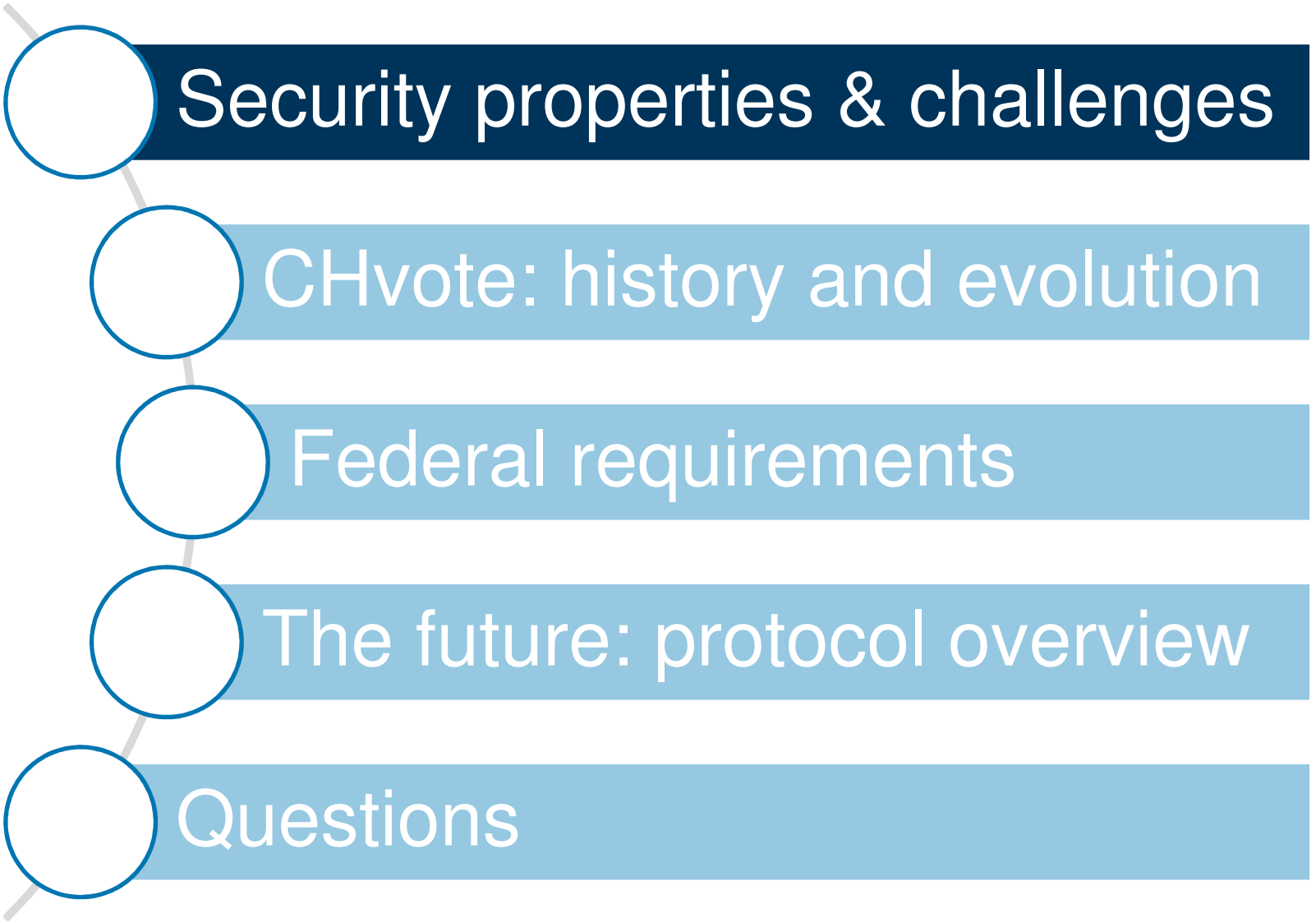
- Now
  - → Internet voting cryptography @ State of Geneva
  - → Java DEV & AppSec

- Outside from work
  - → OWASP-Geneva co-chapter leader
  - → Married, 2 kids

# Outline


Security properties & challenges
CHvote: history and evolution
Federal requirements
The future: protocol overview
Questions

# Outline


Security properties & challenges
CHvote: history and evolution
Federal requirements
The future: protocol overview
Questions

# Security properties

## Target security properties

*Vote secrecy*


*Result integrity*

*Enfranchisement*

*Availability*

*Voter authentication*

*No early tally*

# Security challenges

Partially contradicting requirements and other challenges

- Vote secrecy vs. result integrity
  - → Cryptographically challenging (but feasible)

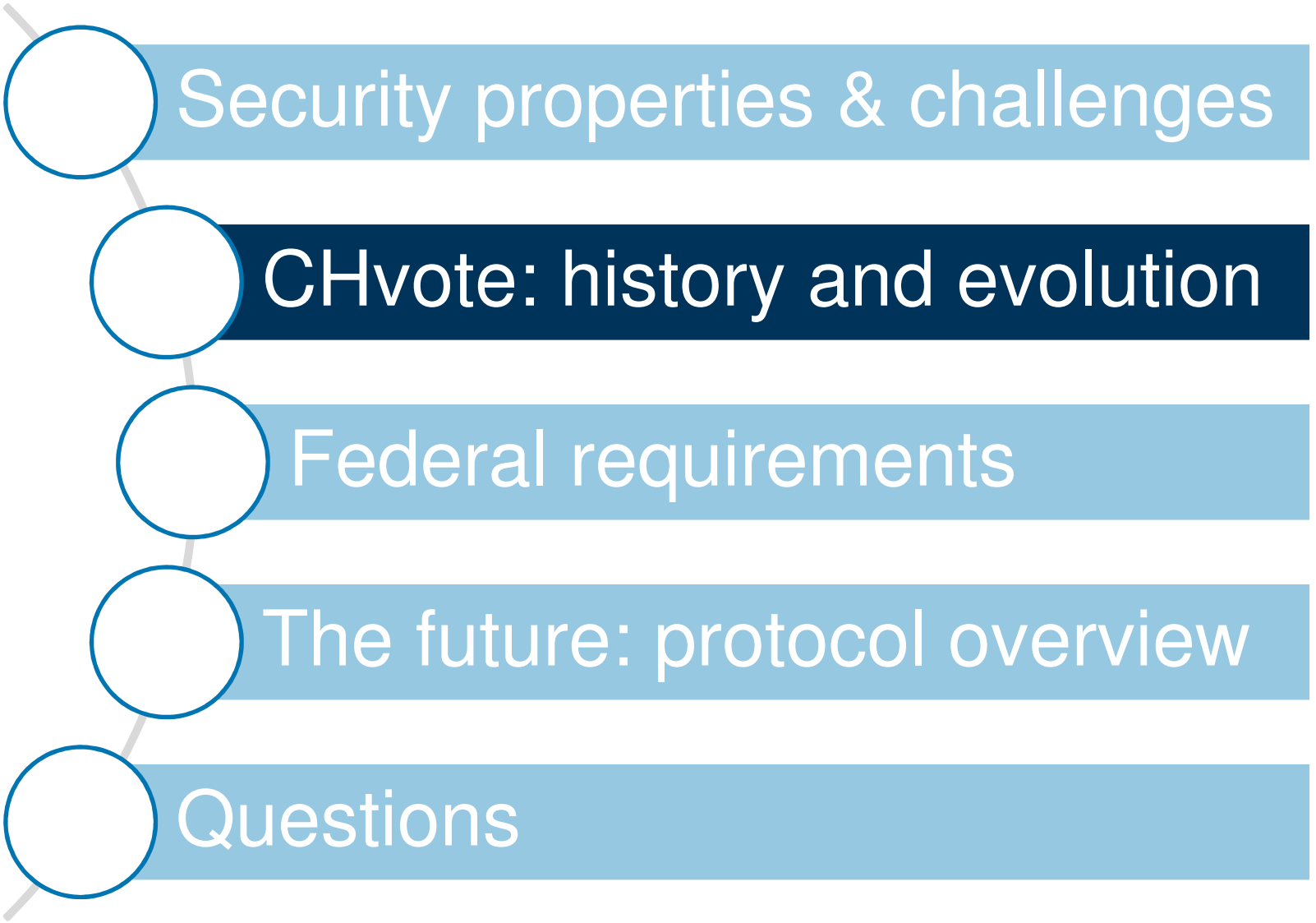
- Enfranchisement vs. authentication
  - → Typically opposed
  - → But: in CH, voting legitimation cards are sent to voters (Swiss Post is trusted)
  - → For mail-in ballots / polling station voting:
    - – Voting card + signature + DOB
  - → For internet voting:
    - – Secrets printed on voting card + DOB

# Security challenges

Partially contradicting requirements and other challenges (ctd.)

- Availability
  - → OK, but… DDOS??
  - → Standard technical counter-measures
  - → Internet voting closes 24 hours before polling stations

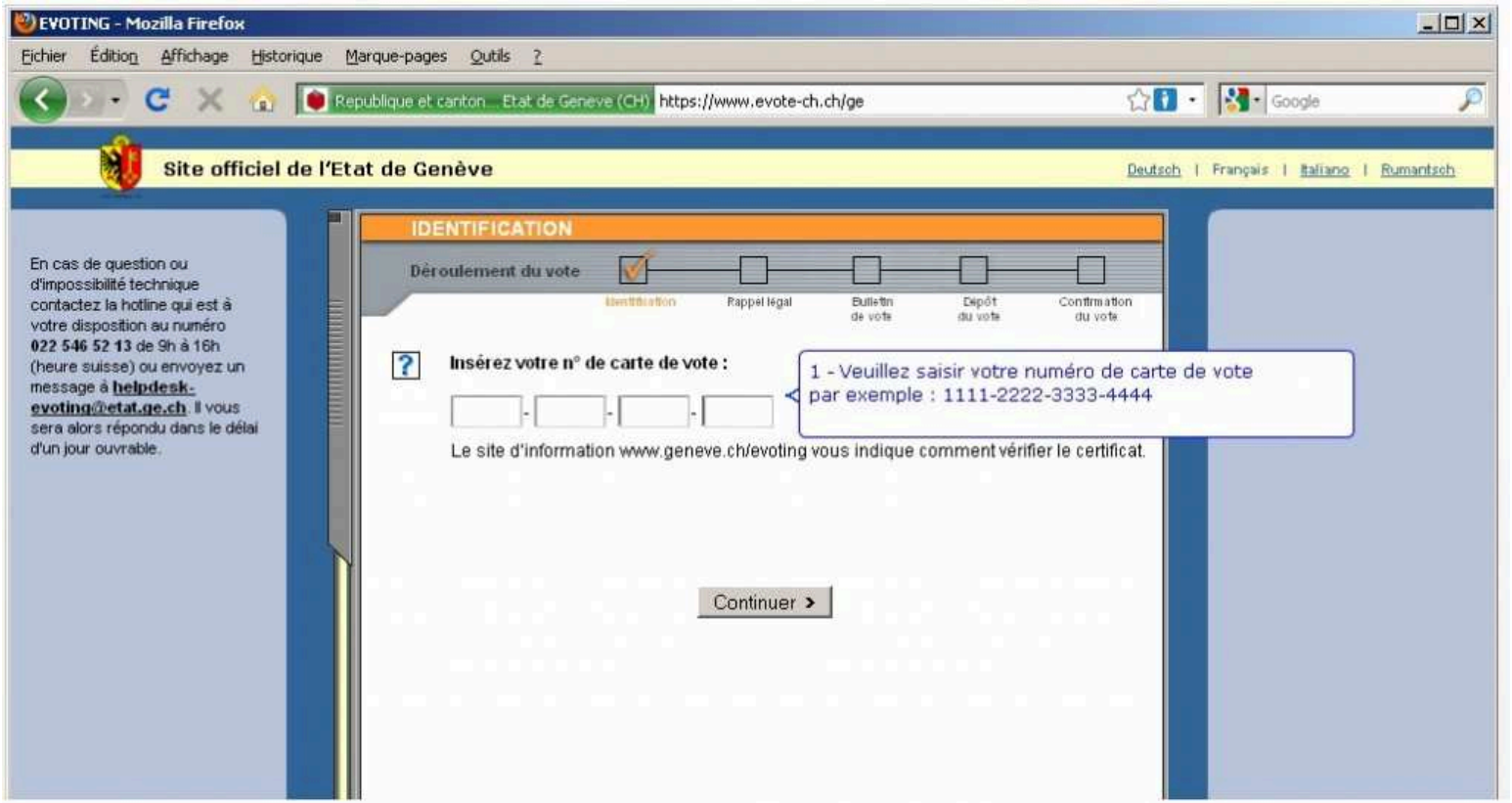
# Outline


Security properties & challenges
CHvote: history and evolution
Federal requirements
The future: protocol overview
Questions

# History

## First generation E-Voting system

- 2001: start of project
- 2003: first use



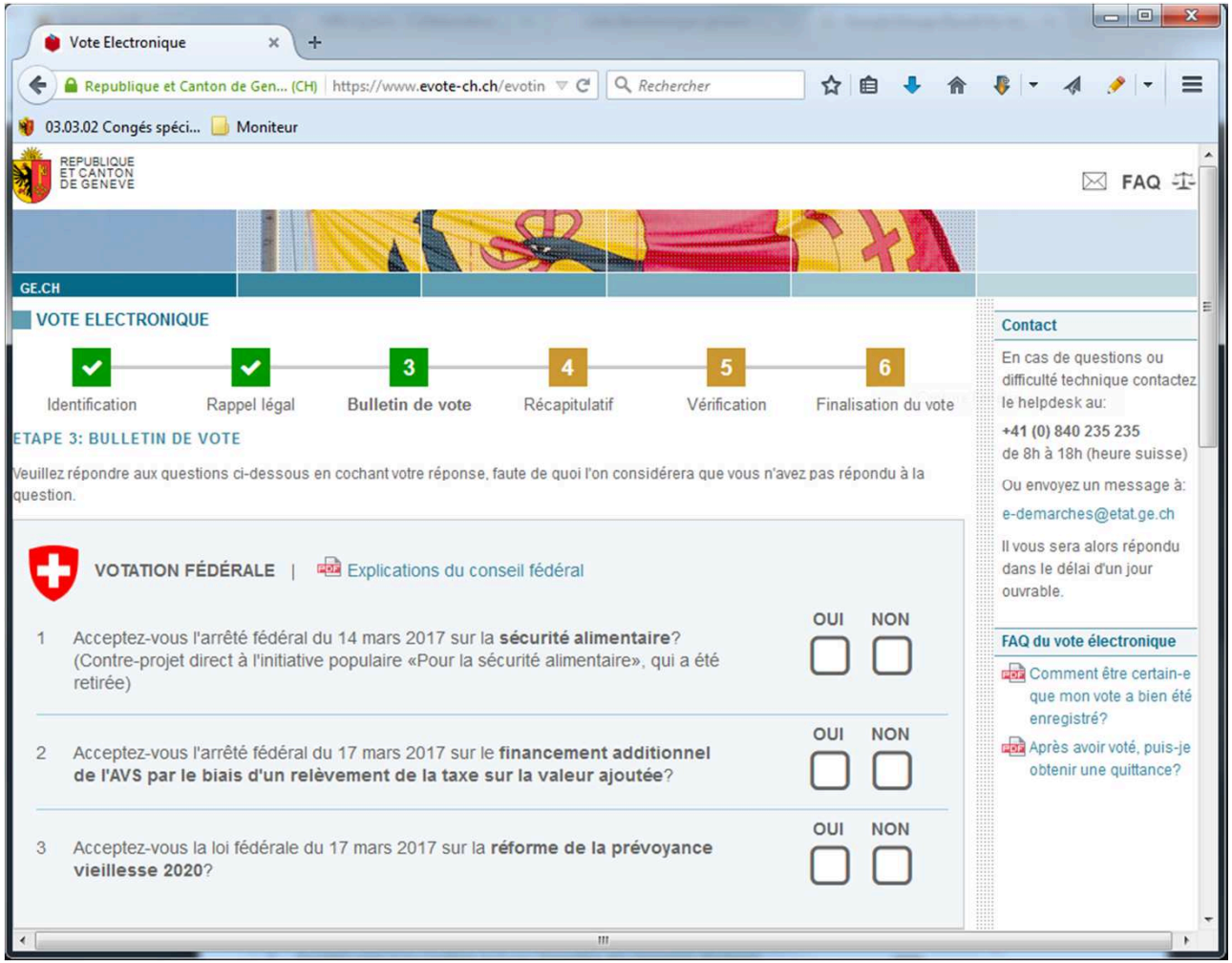
# History

First generation E-Voting system

- 2009: Basel Stadt 
- 2010: Lucern & Bern  

# History

## Updated UI



# History

## New partnerships and future outlook

- Academic partnerships


- New partner cantons

- Opensource and publications

# Evolution

The birth and death of a Java Applet

- Introduced early on
- Additional security layer
  - → Encryption and integrity verification
  - → "Backup" in case of SSL breach
- But
  - → Requires Java on client terminals (and enabled in the browser!)
  - → Prevalent in support requests
  - → Costly in tests (compatibility matrix, jre version, browsers,…)
  - → Delivered over SSL
  - → Imperfect (cf. Nuit du Hack 2013)

# Evolution

The birth and death of a Java Applet

- Removed in 2015

- Individual verifiability
  → "Better" vote integrity verification

# Evolution

Log audits and system monitoring

- Threats made around an important vote
  → Daily **semi-manual** log audits during the voting phase
- Immensely time-consuming
  → No credible threats detected
- Later replaced by in-house specific log auditing tool
- Now using standard SIEM tools, tailored for our needs

# Lessons learnt

"Do not reinvent the wheel!"

- Use standards (when applicable)
  - → More efficient, better understood
  - → Ex.
    - – Splunk vs home made log analyzer
- Do not roll your own applet (or any applet!)
  - → Even more critical for security features

# Lessons learnt

## Partnerships

- Regular audits are mandatory (by law)
  - → Every 3 years, results are public
  - → But insufficient…
- Partnerships with external consulting firms
  - → Several short-term contracts between 2010 and 2011
  - → Stable long term contracts since 2011
  - → Led to improved quality
    - – Design reviews before implementation
    - – Implementation reviews before use for sensitive changes

# Lessons learnt

Partnerships 2.0 – Transparency and open source

- Ethical and idealistic motivations
  - → Democratic processes should be owned by the people…
- TEDx talk by Geneva Chancellor A. Wyden Guelpa
  - → “How can we heal democracy?”
  - → https://www.youtube.com/watch?v=K7gpxZ3FO3s
- But also….

# Lessons learnt

Partnerships 2.0 – Transparency and open source

- *We're better together than alone!*
  - → Already a vulnerability identified and fixed on the protocol
    - – Before it's even live!
- More interesting for academic partners
  - → Open-source means better visibility
- Additional argument for recruitment
  - → Better visibility / CV-boost for developers

# Outline

- Security properties & challenges
- CHvote: history and evolution
- **Federal requirements**
- The future: protocol overview
- Questions

# Federal requirements

New Ordinance on Electronic Voting

- Published in 2013, enacted 2014
  - → Collaborative work between lawmakers, academia and operating staff
- Compliance levels
  - → The higher the compliance, the more voters allowed
- Reference
  - → https://www.bk.admin.ch/themen/pore/evoting/07979/index.html

# Federal requirements

## Individual Verifiability

Voters must receive proof that the server system has registered the vote as it was entered by the voter on the user platform – *VEleS, art. 4*

| **Liste de codes pour la carte n° 5874-8863-1400-8743** | | | |
|---|---|---|---|
| **Votation fédérale** | | | |
| Question 1<br>Acceptez-vous l'arrêté fédéral du 20 juin 2013 portant règlement du financement et de l'aménagement de l'infrastructure ferroviaire (Contre-projet direct à l'initiative populaire "Pour les transports publics", qui a été retirée) ? | Oui<br>A2B4 | Non<br>J5B9 | Blanc<br>Z8H5 |

# Federal requirements

## End-to-End Encryption

Votes must not be stored or transmitted in unencrypted form at any time from being entered to tallying. – *Technical and administrative requirements, section 3.3.4*

# Federal requirements

## Universal Verifiability

For universal verification, the auditors receive proof that the result has been ascertained correctly. They must evaluate the proof in a observable procedure.
– *VEleS, art. 5 paragraph 4*

# Federal requirements

## Control Components

The trustworthy part of the system includes either one or a small number of groups of independent components secured by special measures (control components). Their use must also make any abuse recognisable if per group only one of the control components works correctly and in particular is not manipulated unnoticed. – *VEleS, art. 5, par. 6*

# Federal requirements

## Compliance levels

- First level
  - → Internet voting for up to 30% of voters
- Second level
  - → Add individual verifiability
  - → Add certifying audit
  - → Internet voting for up to 50% of voters
- Third level
  - → Add universal verifiability, control components and end-to-end encryption
  - → New certifying audit
  - → Internet voting for up to 100% of voters

# Outline

- Security properties & challenges
- CHvote: history and evolution
- Federal requirements
- **The future: protocol overview**
- Questions

# Protocol actors

Stakeholders from the perspective of the cryptographic protocol

*Election officer*

*Control components*

*Bulletin Board*

*Printing Authorities*

*Voting client*

*Voter*

# Key cryptographic primitives

A brief overview

- El Gamal homomorphic encryption
- Oblivious Transfer for individual verifiability
  - → Cast-as-Intended Verification in Electronic Elections Based on Oblivious Transfer
- Pedersen Commitments
- Non-interactive Zero-Knowledge Proofs (ZKP)
- Wikström's Proof of a Shuffle

# Homomorphic encryption

What is it?

- Principles
  - → Operations performed on cipher texts
  - → Result visible on recovered plain texts

  - → Example:
    - Encrypt 2
    - Multiply cipher text by 3
    - Decrypt
    - Result is 6

- For this project: El Gamal encryption

# Homomorphic encryption

How and why?

- Used for voter credentials
  - → **Voter authentication**
- Used for encrypting the ballots
  - → **Vote secrecy**
- Allows re-encryptions
  - → Useful for anonymizing when shuffling
  - → **Vote secrecy**
- Allows for key sharing
  - → Control components each hold a key share
  - → **Vote secrecy & result integrity**

# Oblivious Transfer

What does it mean and why is it useful?

- In short
  - → Server knows n secret messages
  - → Client allowed to retrieve k secret messages
  - → Server cannot know which messages the client asked for
  - → *Perfect match for the verification codes issue!*
  - → ***Vote secrecy & Result integrity***

- In detail
  - → Cast-as-Intended Verification in Electronic Elections Based on Oblivious Transfer

# Commitments and ZKPs

How and why?

- “public” commitments for the secrets
  - → Share a value computed from secret, without leaking info
- ZKPs relative to those commitments
  - → Prove that
    - Secret value used in computation =
      secret value used for commitment
  - → Chain of truth from key generation to ballot decryption
- Combination yields Universal verifiability
  - → **Result integrity**

# Wikström's Proof of a Shuffle

Why?

- Re-encrypting mix-net
  - → Each component re-encrypts each ballot and shuffles them

- Since shuffled, simple pre-image proofs would not work
- Since re-encrypted, ciphertexts are not equal
  - → **Vote secrecy**

- Need for a specific proof that the cryptographic shuffle is valid
  - → **Result integrity**

# Outline

- Security properties & challenges
- CHvote: history and evolution
- Federal requirements
- The future: protocol overview
- Questions

# Further reading

And references

- Published protocol specification
  - → https://eprint.iacr.org/2017/325
- Published PoC code
  - → https://github.com/republique-et-canton-de-geneve/chvote-protocol-poc
- Federal requirements
  - → https://www.bk.admin.ch/themen/pore/evoting/07979/index.html

# Thank you!

Thomas Hofer

thomas.hofer@etat.ge.ch

@thhofer