Nový volební systém

Stručný úvod

Každé hlasování má tři druhy stakeholderů, volby mají ještě jeden druh stakeholdera navíc:

Zadavatel

Zadavatel je vždy JEN jeden. Je to ten, který určuje, kdo může hlasovat, může chtít/být nucen toto určení zveřejnit a zahrnout předem danou množinu hlasujících. V případě Svobodných je zadavatelem typicky Republikové předsednictvo (celostátní volby a hlasování politických republikových orgánů), krajské předsednictvo (volby v kraji, hlasování krajských orgánů) či předsedové komisí (hlasování volební, kontrolní či volební komise).

Zadavatel **zná** identitu voličů, může být důvod, aby věděl, kdo hlasoval (a kdo ne). Obvykle nesmí mít možnost zjistit kdo/jak hlasoval. Otevřené hlasování (tj. s informací kdo/jak hlasoval) navržený systém umožňuje.

Volič

Volič je identifikovatelný subjektem. Je nutné zajistit, aby si mohl ověřit správné započítání svého hlasu, a aby jeho hlasování v tajném hlasování skutečně bylo principiálně tajné.

Kandidát

Tento druh stakeholdera se nevyskytuje u hlasování, ale jen u voleb. Je to entita, která má vlastní zájem na ověření férovosti hlasování. Tím, že může sám férovost účinně ověřit, legitimizuje způsob hlasování a pokud není schopen výsledek věrohodně zpochybnit, nezbývá mu, než jej uznat.

Skrutátor (zpracovatel, volební komise)

Skrutátor je ten, kdo počítá hlasy. Obvykle bývá jen jeden „důvěryhodný“, ovšem ve volbách, kde jde o mnoho, je ve skutečnosti skrutátorů více. Běžné volby si totiž teoreticky může nezávisle „spočítat“ každá volební strana, a to prostřednictvím svých zástupců v okrskových volebních komisích. Navržený systém umožňuje, aby volby paralelně sčítal více než jeden systém.

Tento volební systém je navržen tak, aby umožnil:

- ověření započítání hlasu každým hlasujícím voličem,
- zpětné ověření přijetí hlasu každým hlasovacím systémem,
- ověření způsobu provedení hlasování tím, že provedení hlasování úplně probíhá u voliče (standardní mechanismus pomocí HTML/Javascript),
- souběžné zpracování výsledků voleb různými zpracovateli,
- použití anonymizující proxy voličem pro zlepšení anonymity,
- ochranu proti „dosypání hlasů navíc“ vyhlašovatelem hlasování zveřejněním všech hashů vydaných tokenů.

(Ne)rigidita požadavků

Tento dokument předpokládá, že pro komunikaci a podepisování se budou užívat zavedené standardy. Při použití frameworků či implementaci robustního řešení by zřejmě nebyla jiná možnost. V případě jednoúčelového systému však lze XML nahradit např. JSONem a tedy značně zjednodušit i podepisování (tj. bez obecnosti, kterou XML Digital Signature umožňuje). Použití asymetrické kryptografie (RSA, DSA) je však nezbytností. Součástí standardu zveřejněného alespoň pro všechny členy strany musí být úplná specifikace všech komunikačních vazeb vč. použité kryptografie, aby si každý kandidující mohl napsat a použít vlastní systém pro zpracování hlasování.

Implementace na straně zadavatele hlasování

Zadavatel hlasování zná identitu voličů. Jako součást hlasování má vlastní klíč, jehož veřejnou složku vystavuje pro možnost ověření podpisu. Veškeré své výstupy ve vztahu k hlasování elektronicky podepisuje.

Zadání hlasování

Systém před zahájením hlasování musí specifikovat zadání hlasování a zveřejní jej alespoň všem oprávněným zpracovatelům voleb. Zveřejňování se bude provádět voláním ze strany zadavatele, tj. na adresu procesoru s doplněním akce. Akce se doplní jako poslední prvek do URL. Zadání hlasování obsahuje tyto údaje:

- Název hlasování
- Popis hlasování
- Druh hlasování (zaškrtávací, ano/ne)
- Maximální počet zaškrtnutí (jen u zaškrtávacích)
- Začátek hlasování
- Konec hlasování
- Seznam URL zpracovatelů
- Hlasovací možnosti
- Požadované statistické třídění
- Zda má být zveřejněn seznam hlasujících

Zadání hlasování či změna zadání hlasování (možná až do vygenerování tokenů) se provede POST voláním na funkci Ballot, kde xml bude odeslané jako POST proměnná „xml“. Úspěšné zpracování se vždy označí HTTP návratovým stavem 200 OK. Chyba stavem 4xx či 5xx.

V případě, že adresou procesoru je např. *http://vk.svobodni.cz/vote.php*, tedy bude volána adresa *http://vk.svobodni.cz/vote.php/Ballot*

Ukázka zadání (v XML formě, samozřejmě lze zveřejnit i „lidsky čitelné“):

```
<?xml version=”1.0” encoding=”UTF-8”?>
<Ballot xmlns=”http://xsd.svobodni.cz/vs/Ballot-1.0.xsd” Id=”identifikátor”>
   <Name>Primárky pro volby do Evropského Parlamentu 2014</Name>
   <Description>Popis hlasování včetně odkazu na seznam kandidátů</Description>
   <Type>check</Type>
   <MaxChecks>22</MaxChecks>
   <StartTime>2014-02-01 00:00:00</StartTime>
   <EndTime>2014-02-09 23:59:59</EndTime>
   <Processor url=”https://vk.svobodni.cz/vote.php”>Volební komise</Processor>
   <Processor url=”https://volby.mmister.com/hlasuj.php”>Martin Pánek
   (kandidát)</Processor>
   <Option value=”MartinPanek”>Ing. Martin Pánek</Option>
   <Option value=”JiriPayne”>RNDr. Jiří Payne</Option>
   … a další …
   <Stats type=”Region”>Kraj</Stats>
   <Stats type=”Relation”>Druh voliče</Stats> <!-- člen či příznivec? -->
   <ShowVoters>False</ShowVoters>
   <sig:Signature xmlns:sig=”http://www.w3.org/2000/09/xmldsig#”>
      <SignedInfo>…</SignedInfo>
      <SignatureValue>hodnota podpisu</SignatureValue>
      <KeyInfo>
         <KeyValue>…</KeyValue>
         <X509Data>…</X509Data>
      </KeyInfo>
   </sig:Signature>
</Ballot>
```

Vygenerování tokenů a zveřejnění hashů

Podobně zadavatel zveřejní po vygenerování tokenů jejich hashe (pošle je všem zpracovatelským systémům). Podobně jako v případě registrace hlasování vygeneruje XML soubor. Po celou dobu, až do ukončení hlasování, jej může měnit, tedy odebírat hashe a přidávat hashe. To odpovídá zániku volebního práva voličů a vzniku volebního práva voličů v průběhu hlasování. Odstraněním hashem ze seznamu hashů tak zadavatel sděluje zpracovateli, že token odpovídající zmizelému hashi již není platný. Příklad XML s validními hashi (neuvádí se podpisový blok):

```
<?xml version=”1.0” encoding=”UTF-8”?>
<ValidTokens xmlns=”http://xsd.svobodni.cz/vs/ValidTokens-1.0.xsd” Id=”identifikátor”
Ballot=”identifikátor hlasování”>
   <Token>platný hash</Token>
   … a další …
</ValidTokens>
```

Zadavatel hlasování předává oprávněným hlasujícím token (může i jen na vyžádání). Ten obsahuje stejné údaje jako vyhlášení hlasování a k němu navíc specifické údaje, tedy identifikaci tokenu a údaje pro třídění výsledků. Element Ballot je vnořený. Příklad:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<Token xmlns=”http://xsd.svobodni.cz/vs/Token-1.0.xsd” Id=”identifikátor tokenu”>
   <Ballot xmlns=”http://xsd.svobodni.cz/vs/Ballot-1.0.xsd>nepodepsaný ballot</Ballot>
   <Stats type=”Region” id=”11”>Středočeský kraj</Stats>
   <Stats type=”Relation” id=”Member”>Člen strany</Stats>
   <sig:Signature xmlns:sig=”http://www.w3.org/2000/09/xmldsig#”>
   … data podpisu …
   </sig:Signature>
</Token>
```

Zveřejnění výsledků voleb

Není důvod provádět zveřejnění výsledků jinou než lidsky čitelnou formou. Stránka s výsledkem však musí obsahovat u všech možností seznam všech započítaných tokenů. Dále musí obsahovat tabulky výsledků podle statistického třídění (seznam v případě jednoho třídění, tabulku v případě dvou třídění, sadu tabulek v případě více třídění).

Pokud je součástí zadání požadavek na zveřejnění hlasujících, zveřejní se také hashe použitých tokenů.

Implementace na straně zpracovatele (např. volební komise)

Zpracovatel od zadavatele přijímá zadání voleb a seznam hashů. Od voličů přijímá hlasy, které jsou tvořené mj. celým podepsaným tokenem. Jako potvrzení zasílá voličům podepsané hlasovací lístky. Hlasovací lístek opět obsahuje vnořený element Token, u kterého je původní podpis. Hlasovací lístek, tedy hlasovací záznam od voliče samozřejmě není podepsaný. Záznam o hlasování obsahuje navíc jen seznam zvolených možností (u typu „check“ jen označené možnosti s textem „check“, u typu yesno vždy buď „yes“ nebo „no“, nevybrané se vůbec neodesílají). Hlasovací lístek obsahuje také kryptografickou sůl (pseudonáhodný text nebo text zadaný voličem). Příklad odpovědi:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<Vote xmlns=”http://xsd.svobodni.cz/vs/Vote-1.0.xsd” >
   <Token xmlns=”http://xsd.svobodni.cz/vs/Token-1.0.xsd>celý token s podpisem</Token>
   <Option value=”MartinPanek”>check</Option>
   <Salt>TotoJeKryptografickaSul,abyMojeHasheNemohlZadavatelAniOdhadnout</Salt>
</Token>
```

Na takovou hlasovací zprávu zpracovatel v případě úspěchu odpoví potvrzeným hlasovacím lístkem. Ten tedy bude v rámci elementu *Vote* obsahovat také *sig:Signature*.

Přehled metod hlasovacího systému

Veškerá server-side funkčnost vázaná k provozu hlasovacího/volebního systému se váže k systému zpracovatele (způsob práce zadavatele není upravena, na straně volič jde o statickou HTML/JS stránku či jakoukoliv jinou aplikaci podle vůle voliče).

- **Ballot** – zadání/změna zadání voleb (odesílá zadavatel)
- **ValidTokens** – informace o hashích aktuálně platných tokenů (odesílá zadavatel)
- **Vote** – samotné hlasování (odesílá volič)

Implementace volební stránky

Ano, volební stránka, kterou poskytuje volební komise (ovšem kterou volič nemusí využít, pokud si vytvoří či jinak zajistí funkční náhradu), je jen volební komisí poskytována. Standardní volební stránka je HTML s Javascriptovou implementací komunikace s registrovanými volebními systémy. Předpokládá se, že uživatel volební stránce na vstupu poskytne svůj *Ticket*, který obsahuje veškeré informace o hlasování i seznam zpracovatelských systémů. Po provedení volby se kontaktují jednotlivé definované zpracovatelské systémy, které vracejí podepsaný hlasovací lístek. Pokud to prostředí umožňuje, sesumíruje volební systém veškeré odpovědi do textového pole a umožní uživoateli potvrzení vytisknout, zkopírovat přes schránku či uložit na disk (pokud to technologie umožňuje).

Alternativa – JSON

Ballot.JSON

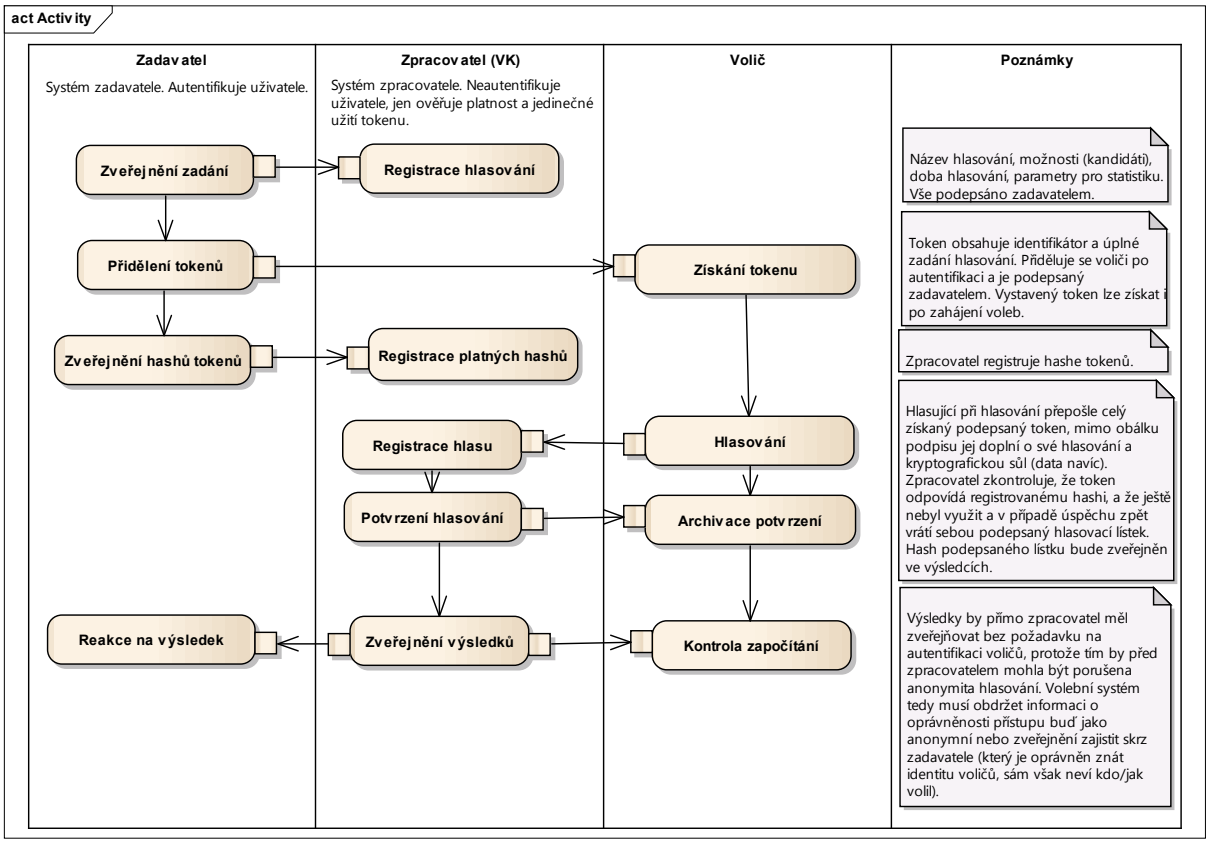
```
{
   “name”             : “Primárky pro volby do Evropského Parlamentu 2014“,
   “description”      : “Popis hlasování včetně odkazu na seznam kandidátů“,
   “type”             : “check”,
   “maxchecks”        : “22”,
   “starttime”        : “2014-02-01 00:00:00”,
   “endtime”          : “2014-02-09 23:59:59”,
   “processor”        : [
      { “url” : “https://vk.svobodni.cz/vote.php”, “name” : “Volební komise” },
      { “url” : “http://www.mmister.com/hlasuj.php”, “name” : “Martin Pánek” }
                        ],
   “option”           : [
      { “value” : “MartinPanek”, “description” : “Ing. Martin Pánek” },
      { “value” : “JiriPayne”, “description” : “RNDr. Jiří Payne” }
                        ],
   … (a tak dál) …
   “sigDigest”        : “hodnotaHashe”,
   “signature”        : “hodnotaPodpisu”,
   “certificate”      : “PEMcertifikát”
}
```

Tak nějak věřím, že z tohoto základu je moje představa asi jasná ☺. Vypadá to složitě, ale je to celé velmi jednoduché.

Diagramy

Activity diagram

Diagram ukazuje činnosti, které provádějí volič, zpracovatel (volební systém) a zadavatel voleb.



Component diagram

Diagram ukazuje vztah systémů zadavatele, zpracovatele a voliče. Kontrolní systém kandidátů je jen dalším systémem zpracovatele voleb.