Dokumentation: Wahlinformationssystem

Tobias Beeh Franziska Geiger Monika Pichlmair

24. Januar 2018

Inhaltsverzeichnis

1. Dokumentation **4**

1.1. Aufsetzen des Systems . . . 4
1.1.1. Datenbank . . . 4
1.1.2. Server . . . 5
1.1.3. Client . . . 5
1.1.4. Vorbereiten einer Wahl . . . 5
1.2. Benutzeroberfläche . . . 6
1.2.1. Auswahl eines Wahljahres . . . 6
1.2.2. Neuaggregierung der Stimmen . . . 6
1.2.3. Parlament . . . 6
1.2.4. District . . . 8
1.2.5. Top10 . . . 9
1.2.6. Difference First Second Votes . . . 10
1.2.7. First Votes . . . 11
1.2.8. Women in Parliament . . . 12
1.3. Berechnung der Ergebnisse . . . 13
1.4. Funktion zur Stimmenabgabe . . . 14
1.4.1. Vorbereitung des Systems für die Stimmenabgabe . . . 14
1.4.2. Wahl . . . 17
1.4.3. Sicherheitskonzept . . . 17
1.5. Benchmarks . . . 18

2. Lastenheft **19**

2.1. Benutzerschnittstellen . . . 19
2.2. Funktionale Anforderungen . . . 19
2.2.1. Ui-Anf-1 . . . 19
2.2.2. Ui-Anf-3 . . . 19
2.2.3. Ui-Anf-4 . . . 20
2.2.4. Ui-Anf-5 . . . 20
2.2.5. Backend-Anf-1 . . . 20
2.2.6. Backend-Anf-2 . . . 20
2.2.7. Backend-Anf-3 . . . 20
2.2.8. Backend-Anf-4 . . . 20
2.2.9. Backend-Anf-5 . . . 20
2.2.10. Backend-Anf-6 . . . 20
2.3. Nicht Funktionale Anforderungen . . . 20
2.3.1. QM-Sicherheit . . . 20

2.3.2. QM-Robustheit . . . 21
2.3.3. QM-Wiederherstellbarkeit . . . 21
2.3.4. QM-Erlernbarkeit . . . 21
2.3.5. QM-Einfachheit . . . 21
2.3.6. RG-1 . . . 21

3. Pflichtenheft **22**
3.1. Zielbestimmung . . . 22
3.1.1. Musskriterien . . . 22
3.1.2. Sollkriterien . . . 22
3.1.3. Kannkriterien . . . 23
3.1.4. Abgrenzungskriterien . . . 23
3.2. Einsatz . . . 23
3.2.1. Anwendungsbereiche . . . 23
3.2.2. Zielgruppen . . . 23
3.3. Umgebung . . . 23
3.3.1. Software . . . 24
3.3.2. Hardware . . . 24
3.4. Funktionalität . . . 24
3.4.1. Anzeige des Wahlergebnisses einer Bundestagswahl . . . 24
3.4.2. Anzeige des Wahlergebnisses einer Bundestagswahl in einem Wahlkreis . . . 25
3.4.3. Abgabe einer Einzelstimme . . . 26
3.4.4. Nichtfunktionale Anforderungen . . . 26
3.4.5. Sicherheit . . . 27
3.4.6. Robustheit und Verfügbarkeit . . . 27
3.4.7. Korrektheit . . . 28
3.4.8. Nutzbarkeit . . . 28
3.4.9. Datengenerator . . . 28
3.5. Daten . . . 28

A. Benchmark-Ergebnisse **30**
A.1. Methodik . . . 30

1. Dokumentation

Im folgenden wird das entwickelte Wahlinformationssystem und die zusätzliche Funktionalität zur elektronischen Stimmabgabe für potenzielle Nutzer des Systems näher erläutert.

1.1. Aufsetzen des Systems

Abhängig von den Sicherheitsanforderungen sind beim Aufsetzen zusätzlich gesonderte Anforderungen zu beachten. Deshalb darf das Aufsetzen eines Produktivsystems grundsätzlich nur von geschulten Technikern durchgeführt werden. Für Test- oder Entwicklungssetups soll die folgende Anleitung jedoch genügen.

1.1.1. Datenbank

Als Datenbank muss ein PostgreSQL-Server, Version 10.1 verwendet werden. Andere Datenbank-Server können funktionieren, erhalten aber keinen Support. Um die Datenbank zu befüllen, muss wie folgt vorgegangen werden:

1. Einrichten des Nutzers `postgres` mit dem Passwort `root` mit Login-Rechten.
2. Erstellen der Datenbank `unicorn`.
3. Einspielen der Datei `election.sql`, beispielsweise mit `psql unicorn postgres < election.sql`. Die Datei beinhaltet das Datenbankschema.
4. Ausführen der Testklasse `BatchInserterTest`. Dadurch wird die Datenbank mit den Wahldaten befüllt.
5. Einspielen der folgenden Dateien:
   a) `aggregation.sql` (initiales Aggregieren der Stimmen)
   b) `insertInhabitantsIntoStates.sql` (Anreicherung der Daten um die Einwohnerzahlen der Bundesländer)
   c) `verteilung.sql` (Erstellen einiger Views zum Berechnen der Sitzverteilung)

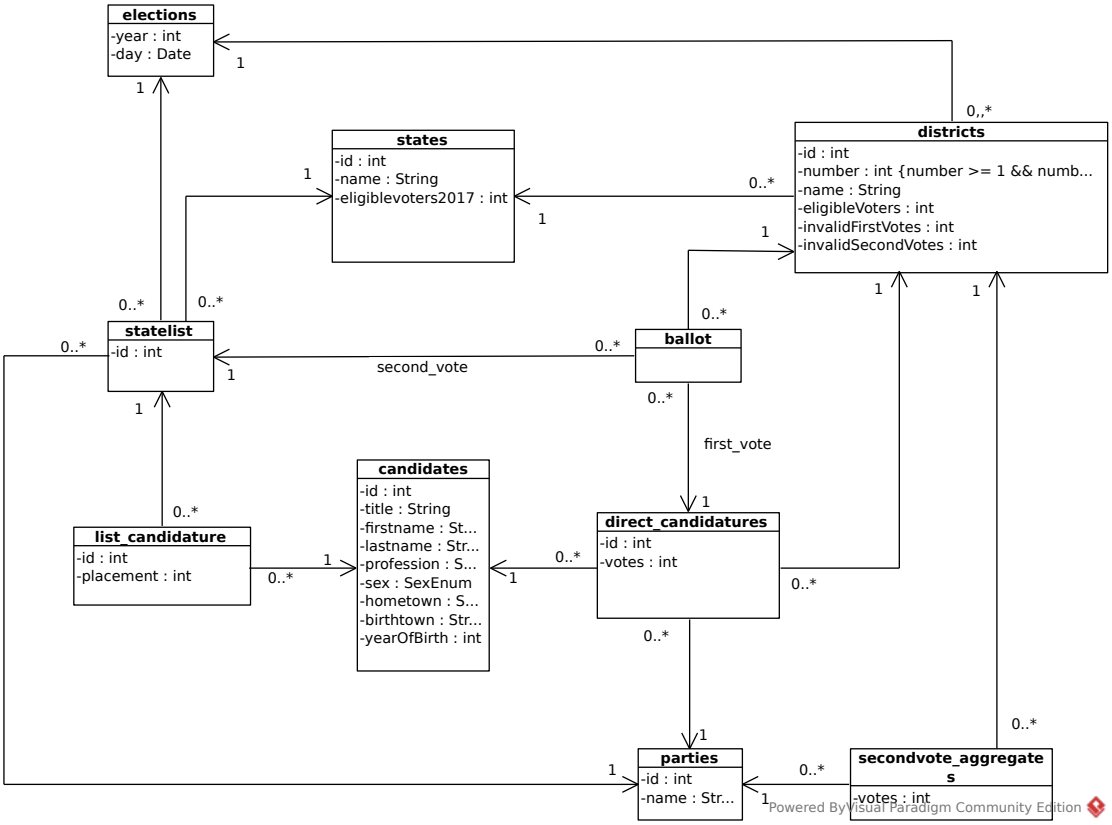


Abbildung 1.1.: Datenbank-Schema

1.1.2. Server

Für Entwicklungszwecke genügt der Server von IntelliJ. Um diesen zu verwenden muss das Projekt in IntelliJ importiert werden, IntelliJ kann mit GWT als Framework von Haus aus umgehen.

1.1.3. Client

Der Client benötigt lediglich einen aktuellen Browser und kann die Anwendung sofort benutzen.

1.1.4. Vorbereiten einer Wahl

Zur Vorbereitung müssen Tokens erzeugt werden. Diese werden mit Hilfe des Java-Projekts `token-gen`, welches im Projektordner liegt, generiert. Die Main-Klasse muss mit den Argumenten `<n> <l1> <l2>` aufgerufen werden, wobei `n` durch die Anzahl der Wahlberechtigten und `l2` durch die benötigte Länge des Tokens zu ersetzen ist. `l1` ist

ein Parameter, mit dem die RAM-Auslastung kontrolliert werden kann [1], der Parameter muss zwischen (jeweils einschließlich) `0` und `n` liegen. Ein höherer Parameter bedeutet hierbei eine geringere RAM-Last. Allerdings kompromittiert ein hoher Parameter die Sicherheit der Tokens. Dabei kann für gegebene Parameter der Sicherheitsverlust wie folgt berechnet werden: Mit $k = \text{floor}(\frac{n}{36^{l1}})$ verringert sich die Sicherheit um $100 \cdot k\%$. Wenn ein Computer mit ausreichend RAM zur Verfügung steht, empfiehlt es sich also, $l1 = 0$ zu wählen.

Das Generieren und Einfügen von 50 Millionen Tokens in die Datenbank dauert je nach Hardware ungefähr 5 bis 10 Minuten.

1.2. Benutzeroberfläche

Die Benutzeroberfläche bietet einem Nutzer die Möglichkeit, sich über die Ergebnisse einer Bundestagswahl zu informieren. Dabei werden die Daten in einem einfachen, verständlichen Format angezeigt

1.2.1. Auswahl eines Wahljahres

Das System kann die Ergebnisse der Bundestagswahlen von 2013 und 2017 anzeigen. Standardmäßig werden die Ergebnisse von 2017 angezeigt. Der Nutzer kann über Checkboxen auswählen, welche Ergebnisse angezeigt werden sollen. Dem System liegen allerdings nur Daten über die Kandidaten von 2017 vor. Deshalb können einige Ergebnisse, wie die prozentuale Anzahl von Frauen im Bundestag, für die Bundestagswahl von 2013 nicht angezeigt werden. In diesen Fällen wird stattdessen eine entsprechende Meldung gezeigt

1.2.2. Neuaggregierung der Stimmen

Um neu eingetragene Stimmen bei den angezeigten Wahlergebnissen zu berücksichtigen kann ein Nutzer die Neuaggregierung der Einzelstimmen über eine Checkbox auslösen. Nach Ende dieser Neuaggregierung werden alle bis zum Zeitpunkt dieser Aggregierung in das System eingetragene Stimmen für die Berechnung der Wahlergebnisse berücksichtigt.

1.2.3. Parlament

Unter 'Parlament' wird die Sitzverteilung im Bundestag angezeigt. Zudem sind dort die prozentuale Verteilung von Stimmen auf Parteien, die Überhangmandate und die Mitglieder des Bundestags zu sehen.

[1] Technischer Hintergrund: Die Tokens werden beim Generieren auf Duplikate geprüft. Da bei vielen Wahlberechtigten und wenig RAM nicht alle Tokens gleichzeitig im RAM vorgehalten werden können, wird die Generierung auf verschiedene Token-Präfixe verteilt. Die Länge des verwendeten Präfixes ist durch diesen Parameter gegeben.

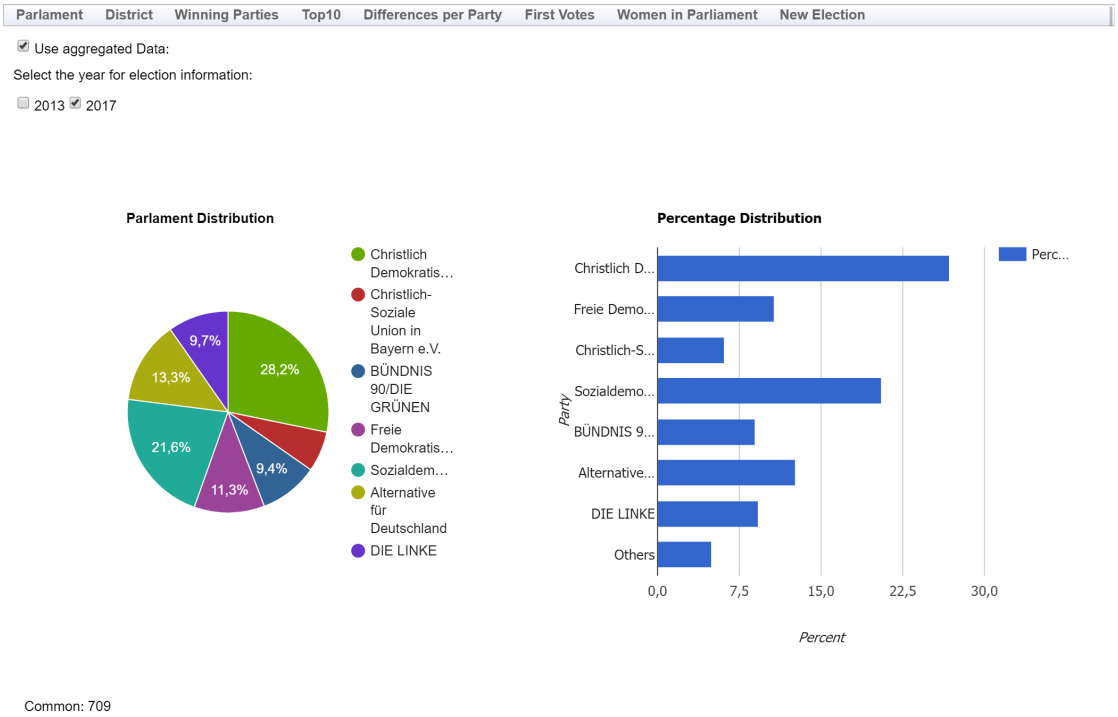


Abbildung 1.2.: Parliament distribution

Parlament Members:

Party	Name	Profession
Alternative für Deutschland	Bernd Baumann	Kaufmann
Alternative für Deutschland	Marc Bernhard	Rechtsanwalt / Geschäftsführer
Alternative für Deutschland	Andreas Bleck	Student
Alternative für Deutschland	Peter Christian Pascal Boehringer	Kaufmann / Wirtschaftspublizist
Alternative für Deutschland	Stephan Brandner	Rechtsanwalt
Alternative für Deutschland	Jürgen Braun	Kommunikationsberater
Alternative für Deutschland	Petr Bystron	Unternehmer
Alternative für Deutschland	Marcus Bühl	Dipl.-Informatiker (M. Sc.) (FH)
Alternative für Deutschland	Matthias Büttner	Informatiker
Alternative für Deutschland	Tino Chrupalla	Malermeister
Alternative für Deutschland	Joana Eleonora Cotar	selbständig
Alternative für Deutschland	Gottfried Curio	MdAB
Alternative für Deutschland	Siegbert Droese	Hotelkaufmann
Alternative für Deutschland	Thomas Ehrhorn	Pilot
Alternative für Deutschland	Gerhard Helmuth Berengar Elsner von Gronow	Vertriebsleiter
Alternative für Deutschland	Michael Espendiller	prom. Mathematiker
Alternative für Deutschland	Peter Felser	Unternehmer
Alternative für Deutschland	Dietmar Friedhoff	Dipl.-Ingenieur / Vertriebstrainer
Alternative für Deutschland	Anton Friesen	wiss. Mitarbeiter

Abbildung 1.3.: Parliament members

AdditionalMandats:

Set Filter:

None

Party	State	Number of additional Mandats
Christlich Demokratische Union Deutschlands	Baden-Württemberg	11
Christlich-Soziale Union in Bayern e.V.	Bayern	7
Christlich Demokratische Union Deutschlands	Sachsen-Anhalt	4
Christlich Demokratische Union Deutschlands	Hessen	3
Christlich Demokratische Union Deutschlands	Brandenburg	3
Christlich Demokratische Union Deutschlands	Sachsen	3
Christlich Demokratische Union Deutschlands	Rheinland-Pfalz	3
Christlich Demokratische Union Deutschlands	Schleswig-Holstein	3
Christlich Demokratische Union Deutschlands	Thüringen	3
Sozialdemokratische Partei Deutschlands	Hamburg	2
Christlich Demokratische Union Deutschlands	Mecklenburg-Vorpommern	2
Christlich Demokratische Union Deutschlands	Saarland	1
Sozialdemokratische Partei Deutschlands	Bremen	1

Abbildung 1.4.: Additional mandats

1.2.4. District

Unter 'District' wird die Anzahl von Wählern und Nichtwählern und die Verteilung von Erst- und Zweitstimmen auf Parteien angezeigt. Diese Daten werden jeweils im Vergleich zu den Daten der vorherigen Bundestagswahl dargestellt. Da das System keine Daten zur Bundestagswahl von 2009 enthält, gibt es hier keine eigene Anzeige für die Bundestagswahl von 2013. Die Ergebnisse von 2013 werden bei den Daten von 2017 als Vergleich mit angezeigt.

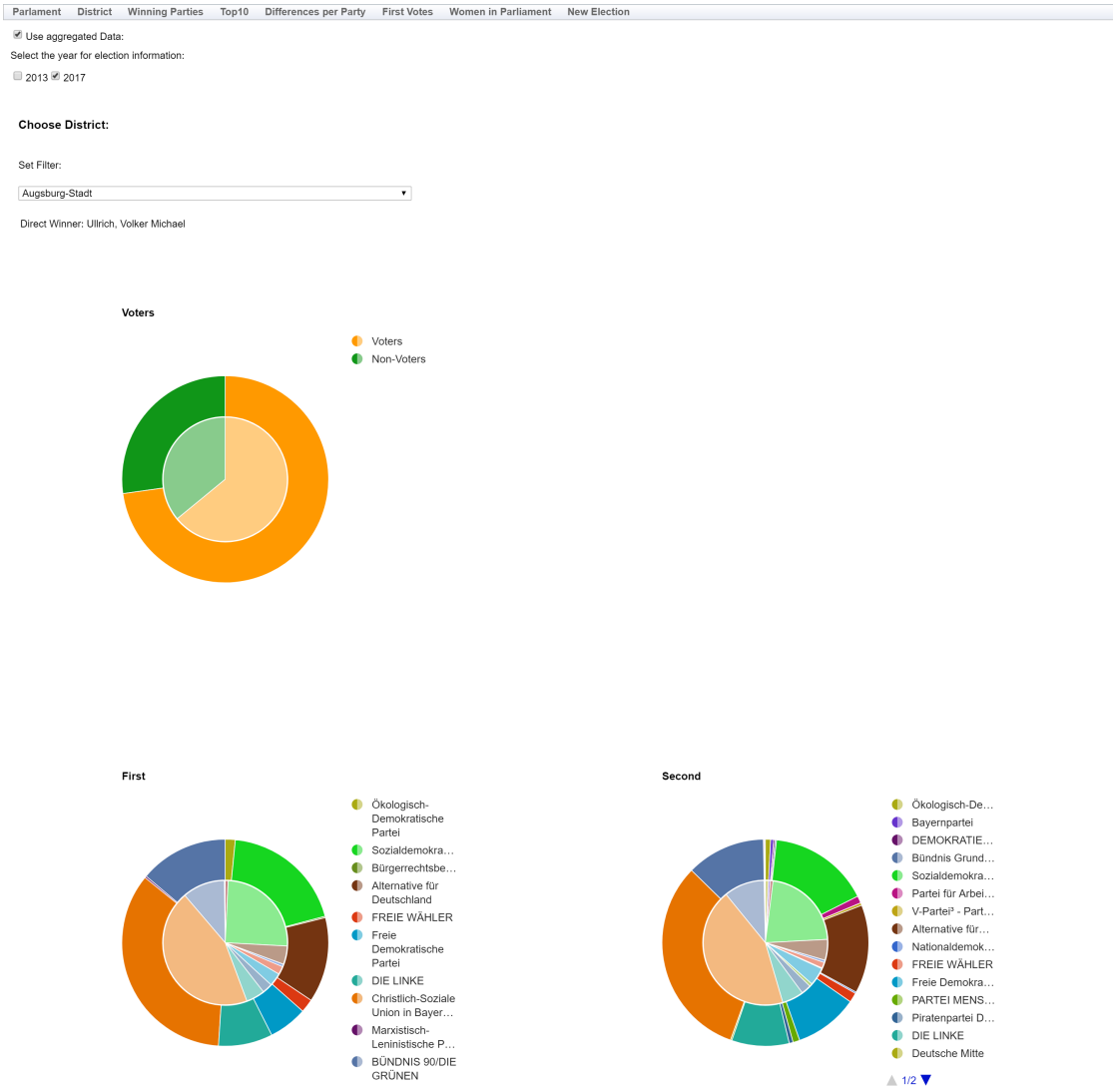
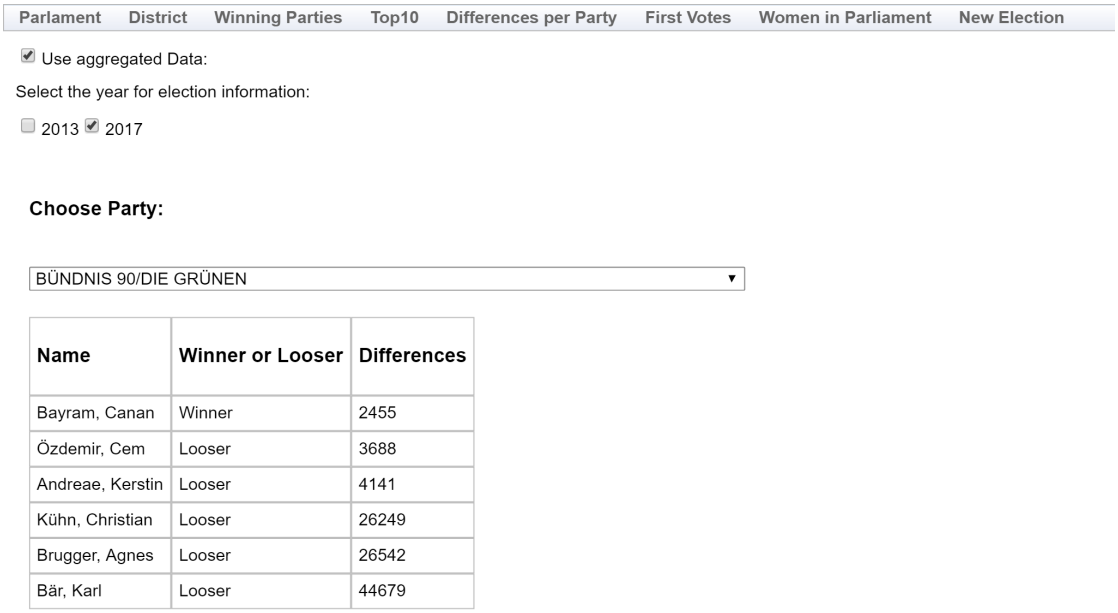


Abbildung 1.5.: District view

1.2.5. Top10

Unter 'Top 10' werden – pro Partei – die knappsten Gewinner und Verlierer bezüglich der Erststimmen angezeigt. Hierbei werden die Gewinner und die „Verlierer", also die zweitplatzierten Kandidaten, berücksichtigt. Angezeigt werden solche Kandidaten, die bundesweit am knappsten gegen die Konkurrenz gewonnen oder verloren haben. Diese Information ist besonders für die Parteien interessant: In diesen Wahlkreisen lohnt sich der Wahlkampf am meisten.



Name	Winner or Looser	Differences
Bayram, Canan	Winner	2455
Özdemir, Cem	Looser	3688
Andreae, Kerstin	Looser	4141
Kühn, Christian	Looser	26249
Brugger, Agnes	Looser	26542
Bär, Karl	Looser	44679

Abbildung 1.6.: Top 10 view

1.2.6. Difference First Second Votes

Unter 'Difference First Second Votes' wird für jede Partei einer Bundestagswahl angezeigt, in welchem Wahlkreis der Abstand zwischen ihrer Erst- und Zweitstimmen am größten war. Die Größe dieses Abstands wird als Balkendiagramm angezeigt. Wenn der Nutzer mit der Maus über eine dieser Balken fährt kann er sehen, in welchem Wahlkreis der Abstand zwischen den Erst- und Zweitstimmen der Partei am größten war und ob sie mehr Erst- oder mehr Zweitstimmen erhalten haben. Daran kann man sehen, ob ein bestimmter Erstkandidat besonders beliebt oder besonders unbeliebt war, oder ob der Erstkandidat wenig Einfluss auf die Anzahl der erhaltenen Erststimmen hatte. Zudem kann geprüft werden, ob an sehr kleine oder sehr große Parteien bevorzugt Erst- oder Zweitstimmen vergeben werden.

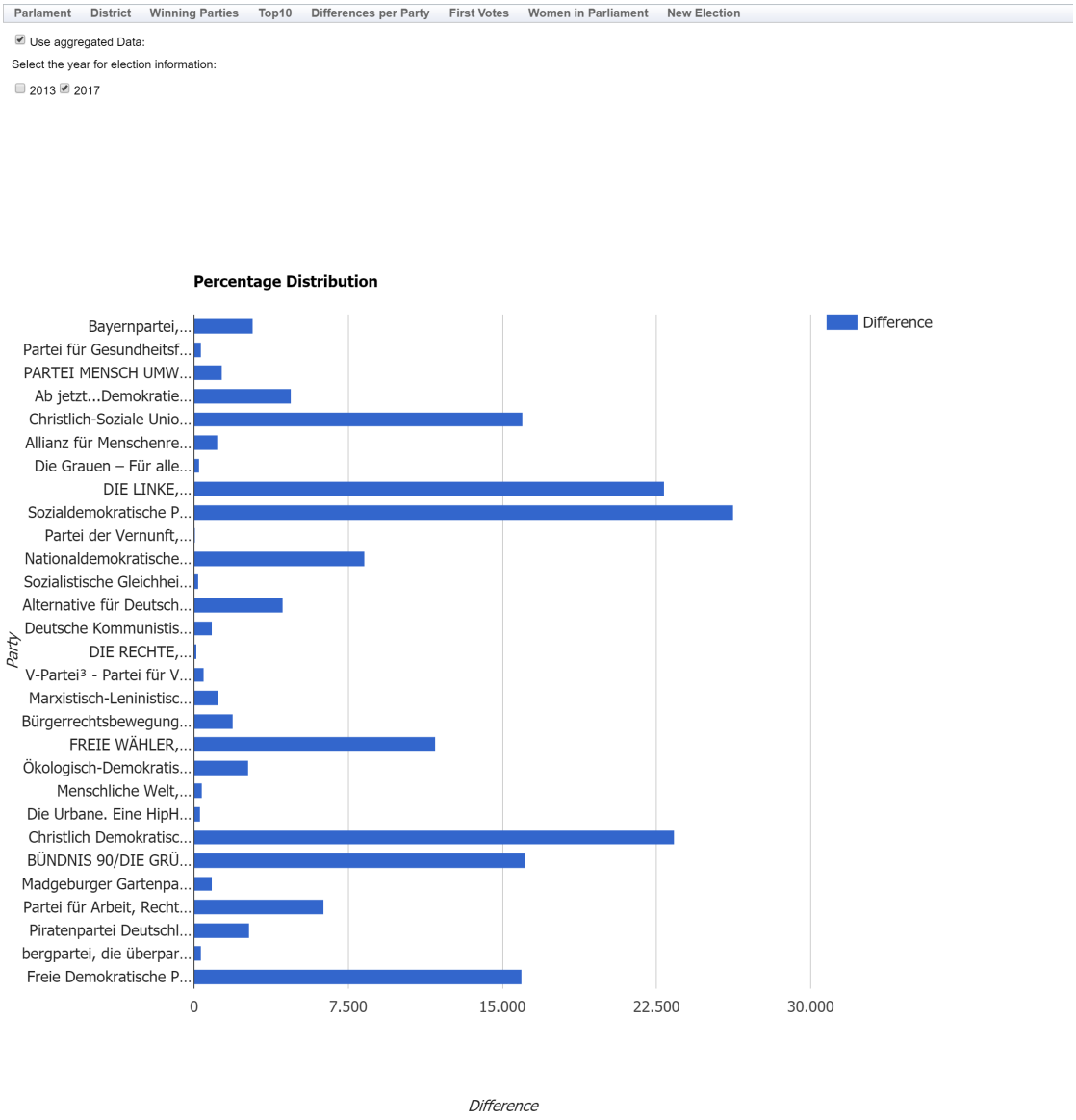


Abbildung 1.7.: Differences per party

1.2.7. First Votes

Unter 'First Votes' wird die Verteilung der Erststimmen auf Bundestagsebene angezeigt. Hier kann der Nutzer sehen, ob eine stärkere Gewichtung der Erststimmen das Ergebnis einer Bundestagswahl beeinflussen könnte.

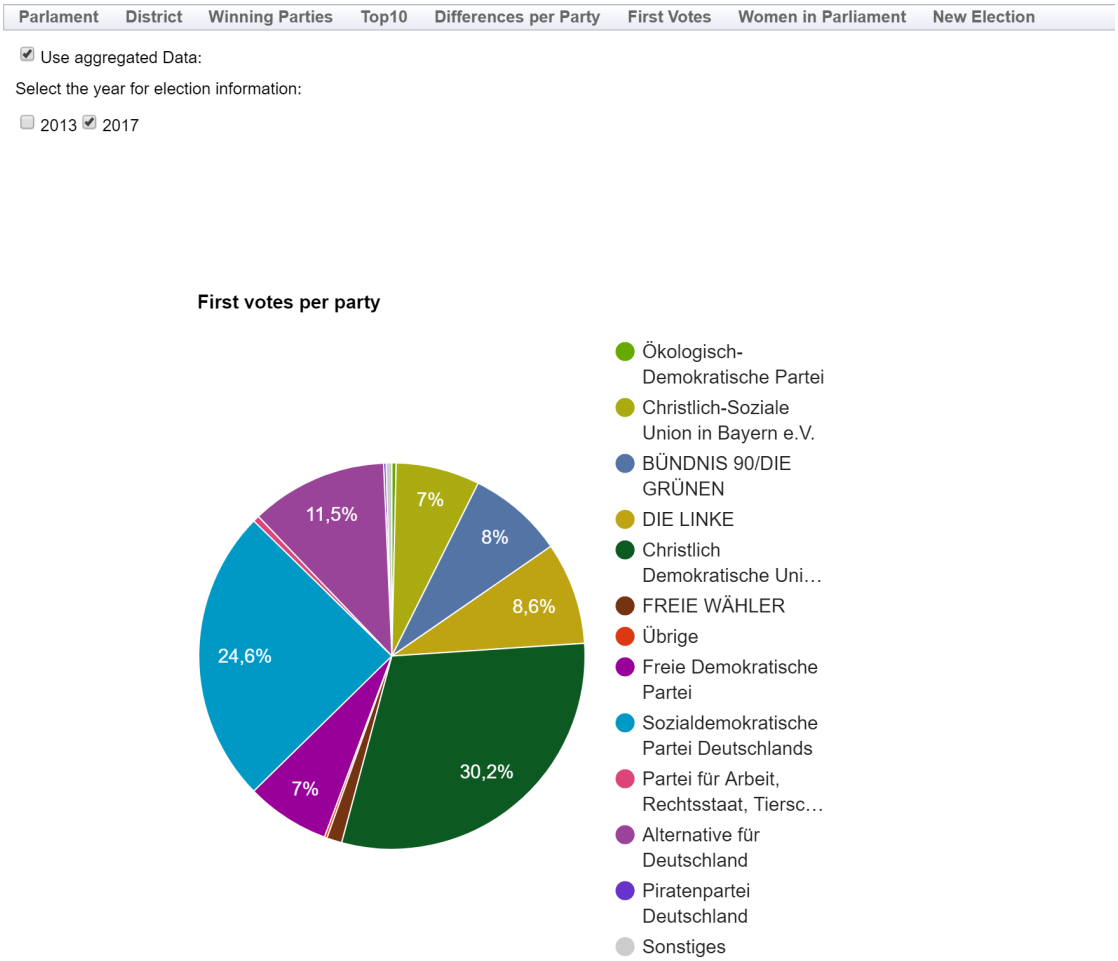


Abbildung 1.8.: First votes view

1.2.8. Women in Parliament

Unter 'Women in Parliament' wird die prozentuale Anzahl der Frauen und Männer im Bundestag angezeigt. Da das System keine entsprechenden Daten über die Kandidaten von 2013 enthält ist diese Anzeige nur für die Bundestagswahl von 2017 möglich

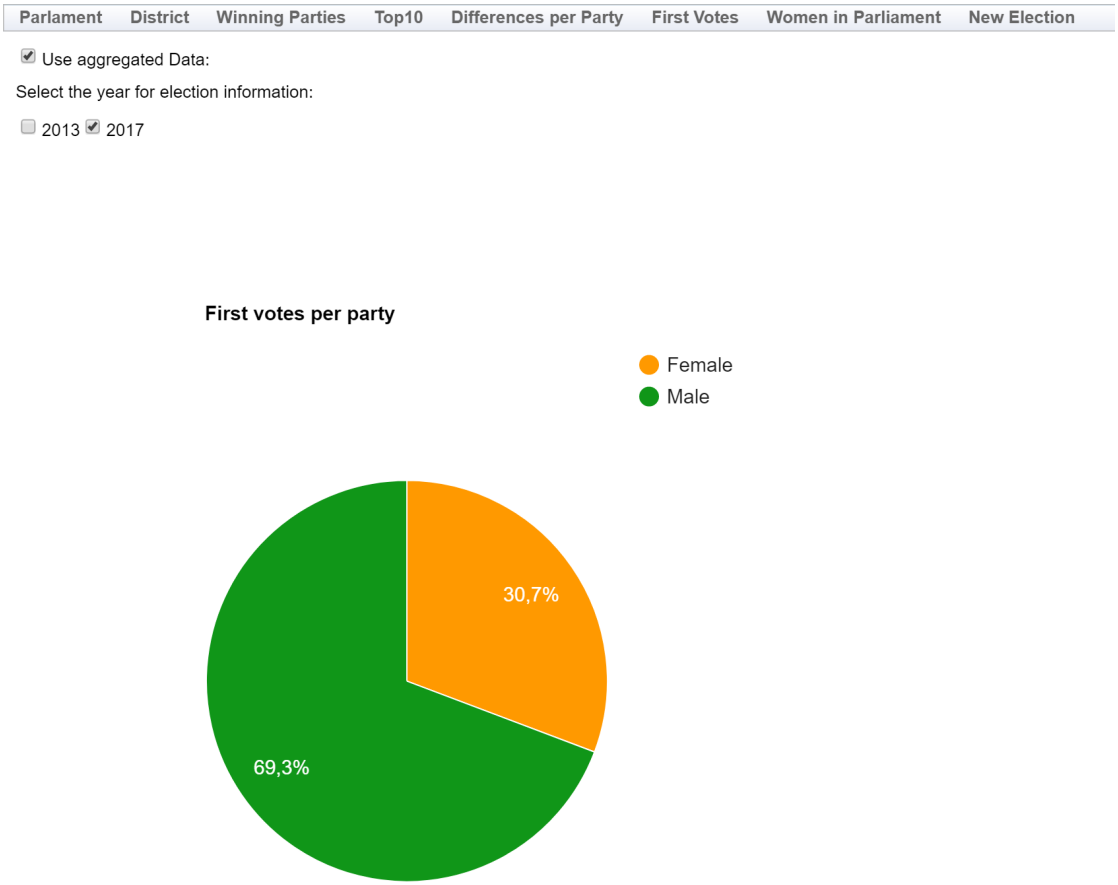


Abbildung 1.9.: Women in parliament view

1.3. Berechnung der Ergebnisse

Als Grundlage für die Berechnung dienen die Stimmverteilungen auf Wahlkreisebene, die auf https://www.bundeswahlleiter.de/ einsehbar sind. Hierbei werden für 2013 die Ergebnisse verwendet, die bei den Wahlergebnissen von 2017 enthalten sind. Diese Ergebnisse wurden auf die neu verteilten Wahlkreise für die Bundestagswahl von 2017 umgerechnet.

Die Berechnung der Ergebnisse der Bundestagswahlen, die von dem Wahlinformationssystem angezeigt werden, erfolgen ausschließlich über SQL-Anfragen. Eine kommentierte Version der verwendeten Anfragen ist dieser Dokumentation beigefügt.

1.4. Funktion zur Stimmenabgabe

Das Wahlinformationssystem unterstützt eine Stimmabgabe-Funktion. Dies ermöglicht es, ein Wahllokal auf digitale Wahl umzustellen, womit der Prozess der Stimmenauszählung automatisiert erfolgen kann.

1.4.1. Vorbereitung des Systems für die Stimmenabgabe

Zunächst müssen für eine Wahl Tokens erstellt werden. Dies geschieht mit einem Kommandozeilenprogramm im Vorfeld. Die Tokens werden daraufhin in der benötigten Menge für jedes Wahllokal gedruckt.

Im Wahllokal werden nun ein oder mehrere Wahlcomputer aufgebaut. Diese müssen eine stabile Verbindung zum zentralen Backend-Server aufweisen.

In einem Browser wird nun die Weboberfläche des Wahlsystems geöffnet. Auf der Website wird die Funktion 'New Election' gewählt. Daraufhin wird eine Auswahlseite angezeigt, in der der Wahlkreis ausgewählt wird.

Abbildung 1.10.: District selection view

Nach Auswahl eines Wahlkreises wird eine Eingabemaske gezeigt, in der ein Wähler später ein Wahltoken eingeben kann. Nach Eingabe eines gültigen Tokens wird der Stimmzettel angezeigt, auf dem der Wähler wählen kann. Die wählbaren Direktkandidaten und Landeslisten werden dabei in der gleichen Reihenfolge angezeigt, die sie auch auf einem konventionellen Stimmzettel haben würden. Dabei orientiert sich die Reihenfolge der Parteien an der Anzahl der Zweitstimmen, die sie bei der vorherigen Bundestagswahl in dem entsprechenden Bundesland des Wahlkreises erhalten haben.

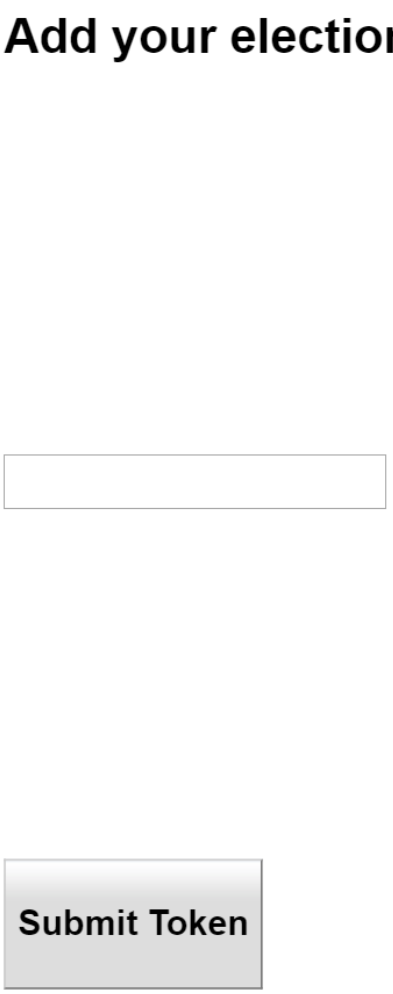


Abbildung 1.11.: Token input view

Ballot

First Vote

Range	Candidate		Check
1	**Norbert Brackmann**	MdB / Jurist **Christlich Demokratische Union Deutschlands**	
2	**Dr. Nina Scheer**	Juristin / Politikwissenschaftlerin / Musikerin **Sozialdemokratische Partei Deutschlands**	
3	**Dr. Bernd Buchholz**	Rechtsanwalt **Freie Demokratische Partei**	
4	**Dr. Konstantin von Notz**	MdB / Jurist **BÜNDNIS 90/DIE GRÜNEN**	
5	**Dr. Bruno Hollnagel**	Wirtschaftswissenschaftler **Alternative für Deutschland**	
6	**Heidi Herma Elise Beutin**	Wissenschaftspublizistin **DIE LINKE**	
7			
8	**Gregor Voht**	selbst. Kaufmann **FREIE WÄHLER**	
9			
10			
11			
12			

Second Vote

Check	Party		Range
	Christlich Demokratische Union Deutschlands	Johann David Wadephul Astrid Damerow Ingo Gädechens Gero Storjohann Norbert Brackmann	1
	Sozialdemokratische Partei Deutschlands	Ernst Dieter Rossmann Gabriele Hiller-Ohm Nina Scheer Bettina Hagedorn Sönke Rix	2
	Freie Demokratische Partei	Wolfgang Kubicki Gyde Jensen Bernd Buchholz Tobias Mährlein Christine Aschenberg-Dugnus	3
	BÜNDNIS 90/DIE GRÜNEN	Luise Amtsberg Friederike Löffert-Pokatis Ingrid Nestle Jörg Nickel Konstantin von Notz	4
	Alternative für Deutschland	Gereon Martin Bollmann Bruno Hollnagel Axel Gehrke Dennis Wamhoff Joachim Schneider	5
	DIE LINKE	Sascha Luetkens Lorenz Gösta Beutin Ayse Fehimli-Kuzu Cornelia Möhring Katjana Zunft	6
	Partei für Arbeit, Rechtsstaat, Tierschutz, Elitenförderung und basisdemokratische Initiative	Clemens Fonk Swana-Aluna Lohse Jannis Langmaack André Mevius Bastian Langbehn	7
	FREIE WÄHLER	Monica Farell Daniel Biedermann Gregor Voht Rainer Schuchardt Thomas Misch	8
	Bündnis Grundeinkommen	Susanne Fritz Ines Thönnißen Holger Thiesen Malte Kanthack Anton Robert Doll	9
	Nationaldemokratische Partei Deutschlands	Ingo Stawitz Rudolf Rosenthal Jörn Lemke Mark Michael Proch Wolfgang Schimmel	10
	Ökologisch-Demokratische Partei	Georg Siebert Jörg Petrulat Ulrich Otto Niedermann Michael Möller Elke Kathleen Trede	11
	Marxistisch-Leninistische Partei Deutschlands	Lüder Möller Helin Kaya Eike Wilke Maria Meyer Karin Zan Bi	12

Submit Votes

Abbildung 1.12.: Ballot view

Aus Sicherheitsgründen muss die Website auf dem Wahlcomputer in den Vollbildmodus versetzt werden. Der Wähler darf keine Möglichkeit haben, diesen Vollbildmodus zu beenden. So kann sichergestellt werden, dass der Wähler ausschließlich die vorbereitete Eingabemaske benutzen kann.

1.4.2. Wahl

Ein Wähler kommt in das Wahllokal mit seinem Personalausweis und seiner Wahlberechtigung. Die Wahlhelfer registrieren den Wähler konventionell. Nun darf der Wähler ein Zettel mit einem Wahltoken nehmen. Hierbei stellen die Wahlhelfer sicher, dass jeder Wähler genau ein Token nimmt.

Nun darf der Wähler zum Terminal gehen, sein Token in die Maske eintragen und seine Wahl abgeben. Dabei kann die Stimme manuell ungültig gemacht werden, indem etwa für die Erst- oder Zweitstimme eine leere Zeile oder gar keine Zeile ausgewählt wird. Anschließend bestätigt der Wähler seine Eingabe. Nachdem der Wähler seine Stimme so abgegeben hat wird wieder die initiale Eingabemaske angezeigt, sodass der nächste Wähler wählen kann.

1.4.3. Sicherheitskonzept

Tokens

Damit die Tokens gegen Brute-Force Attacken geschützt sind, müssen die Tokens eine bestimmte Entropie aufweisen. Hierbei gilt: Die Anzahl der *möglichen* Tokens dividiert durch die Anzahl der *benötigten* Tokens darf 2^{40} nicht unterschreiten.

Hierzu genügt aktuell (ca. 50 Millionen Wähler) beispielsweise ein alphanumerisches, case-insensitives Token mit 16 Zeichen (der Rest-Puffer wäre hierbei ca. 140, das heißt dass wir die Anzahl der Wähler noch um Faktor 140 vergrößern könnten, bevor das Token zu kurz wird). Zusätzlich wird im System jeder Wahlversuch mit einem falschen Token geloggt, sodass Brute-Force Angriffe rückwirkend nachweisbar wären.

Absicherung des Systems

Zur weiteren Absicherung gegen Brute-Force-Attacken wird der öffentliche Zugriff auf das produktive Wahlsystem für die Dauer der Wahl entzogen. Die Wahlergebnisse dürfen ohnehin nicht vor dem Ende der Wahl veröffentlicht werden. Des Weiteren kann eine Kopie des Systems weiterhin öffentlich erreichbar sein. Das produktive Wahlsystem wird hierbei in einem privaten Netzwerk betrieben, wobei der Zugriff von den Wahlcomputern via VPN stattfindet.

Absicherung des Wahlcomputers

Die Weboberfläche auf jedem Wahlcomputer muss den Zugriff nach 10 konsekutiven Fehlversuchen für die Eingabe des Tokens sperren und kann nur durch einen Wahlhelfer, der den Vollbildmodus beenden und die Seite neu laden kann, wieder entsperrt werden. So wird zusätzlich das zufällige Ausprobieren von Zeichenketten verhindert.

1.5. Benchmarks

Die Performance sowie die Ressourcenauslastung des Systems wurde mit Apache JMeter[2] getestet. Da das verwendete Framework, GWT, zwischen Frontend und Backend mit authentifizierten RPCs kommuniziert, haben wir separat ein REST-Interface programmiert. Die durchgeführten Berechnungen im Backend sind dabei die selben wie beim Aufruf einer Seite des Frontends.

Die JMeter-Konfiguration, die dem Projekt beiliegt, sendet zufällige Anfragen an den Server. Dabei ist die Parallelität, also die Anzahl der simulierten User, sowie die Verzögerung zwischen zwei Anfragen eines Users frei konfigurierbar.

Da uns zum Entwicklungszeitpunkt leider kein potenter Server zur Verfügung steht, können wir keine Hardwareempfehlungen für die Software geben. Jedoch konnten wir in den Benchmarks feststellen, dass die Performance des Backends bei zusätzlichen Nutzern linear skaliert, bis alle Ressourcen des Servers aufgebraucht sind. Dies bedeutet, dass sich die Anwendung mit entsprechenden Servern nahezu beliebig skalieren lässt.

Ein zusätzliches Caching der Anfragen könnte leicht in die Anwendung eingebaut werden und würde zu einer wesehtlich höheren Skalierbarkeit führen, da die Anwendung zusätzliche, parallele Anfragen dann in vernachlässigbarer Zeit durchführen könnte.

Die Ergebnisse der Benchmarks befinden sich im Anhang A.

[2] `https://jmeter.apache.org/`

2. Lastenheft

Im Folgenden werden die Anforderungen an das Wahlinformationssystem knapp zusammengefasst dargestellt.

2.1. Benutzerschnittstellen

Folgende Aktionen sollen für einen Nutzer des Wahlinformationssystems möglich sein:

- Abfrage des Ergebnisses der Bundestagswahl von 2013 oder 2017, bestehend aus der Sitzverteilung im Bundestag, der Überhangmandate und der Mitglieder des Bundestags
- Abfrage der Ergebnisse der Bundestagswahl eines beliebigen Wahlkreises, bestehend aus der Wahlbeteiligung, des gewählten Direktkandidaten, der Anzahl an Erst- und Zweitstimmen und die für jede Partei abgegeben wurden. Die Entwicklung der Verteilung der Stimmen im Vergleich zum Vorjahr soll hier sichtbar sein.
- Abfrage einer Auflistung aller Wahlkreise und der Parteien, die in diesem Wahlkreis die meisten Erst- oder Zweitstimmen erhalten haben.
- Die Abfrage der Kandidaten einer Partei, die in ihrem Wahlkreis besonders knapp gewonnen oder verloren haben.

2.2. Funktionale Anforderungen

2.2.1. Ui-Anf-1

Das System muss dem Nutzer die Möglichkeit bieten verschiedene Wahljahre auszuwählen. Alle Abfragen müssen für die Bundestagswahlen von 2017 und, soweit entsprechende Daten zur Verfügung stehen, 2013 möglich sein.

2.2.2. Ui-Anf-3

Das System muss dem Nutzer eine Übersicht der gesamten Wahldaten in aufbereiteter Form einzusehen. Die Wahldaten müssen in Form von Charts und Diagrammen visualisiert und verständlich dargestellt werden. Ein unbeteiligter Nutzer muss in der Lage sein, die Sicht ohne zusätzliche Anleitungen zu verstehen.

2.2.3. Ui-Anf-4

Das System muss dem Nutzer die Möglichkeit bieten die Ergebnisse einer Wahl für einen bestimmten Wahlkreis anzuzeigen.

2.2.4. Ui-Anf-5

Die Applikation muss in einem Browser ausgeführt werden können. Durch Eingabe einer URL im Browser muss sich die Applikation öffnen und nutzbar sein.

2.2.5. Backend-Anf-1

Das System muss eingetragene Daten auch nach Neustart noch laden können. Nach dem Neustart der Software muss das selbe Wahlergebnis zu den jeweiligen Wahlen sichtbar sein wie vor dem Neustart.

2.2.6. Backend-Anf-2

Das System muss die Berechnung der Wahlergebnisse unter Berücksichtigung von bestimmten Sonderklauseln, wie etwa der 5-Prozent Hürde durchführen können. In der Gesamtübersicht der Stimmenverteilung werden Randparteien unter einem aggregierten Punkt dargestellt.

2.2.7. Backend-Anf-3

Das System muss anhand der eingetragenen Stimmen nach dem aktuellen Wahlverfahren die Sitzverteilung des Bundestags berechnen können.

2.2.8. Backend-Anf-4

Das System soll die Wahlbeteiligung für einen ausgewählten Wahlkreis anzeigen.

2.2.9. Backend-Anf-5

Das System muss die Wahldaten in einer Datenbank persistieren.

2.2.10. Backend-Anf-6

Das System muss Überhangmandate in der Berechnung der Sitzverteilung im Bundestag mit einbeziehen und darstellen können.

2.3. Nicht Funktionale Anforderungen

2.3.1. QM-Sicherheit

Das System muss den heutigen Sicherheitsstandards entsprechen. Die Stimmen dürfen nur anonymisiert abgespeichert werden. Stimmdaten dürfen nicht mit persönlichen Daten

eines Wählers in Verbindung gebracht werden können.

2.3.2. QM-Robustheit

Das System arbeitet auch unter sehr hoher Last fehlerfrei. Eine sehr hohe Anzahl von Anfragen darf die Daten des Systems in keiner Weise beeinflussen. Der Nutzer soll weiterhin nur eine vertretbare Wartezeit haben.

2.3.3. QM-Wiederherstellbarkeit

Das System muss nach einem Neustart den selben validen Zustand wie zuvor haben. Die angezeigten Daten dürfen sich nach einem Neustart nicht verändert haben.

2.3.4. QM-Erlernbarkeit

Der Nutzer muss die Applikation ohne Zuhilfenahme einer Dokumentation verstehen. Ein Unbeteiligter muss nach 15 Minuten fähig sein mit der Software umzugehen.

2.3.5. QM-Einfachheit

Die Benutzerschnittstellen sollen möglichst einfach gestaltet werden und keine überflüssigen Informationen beinhalten. Die Sichten der Applikation sollen nicht überladen wirken, aber trotzdem alle wichtigen Informationen beinhalten.

2.3.6. RG-1

Die Software wird in der Programmiersprache Java verfasst, nutzt zur Oberflächengenerierung GWT und arbeitet mit einer PostgreSQL Datenbank zusammen.

3. Pflichtenheft

In diesem Projekt soll ein Wahlinformationssystem entwickelt werden. Das System soll einem Nutzer die Ergebnisse der aktuellen oder einer vorherigen Bundestagswahl anzeigen können. Der Nutzer soll sich mit Hilfe des Systems unter anderem über die Stimmenverteilung einer Wahl informieren können. Diese Daten sollen nicht nur für die gesamte Wahl, sondern auch auf Wahlkreisebene angezeigt werden können. Das System soll für die elektronische Abgabe von Stimmen genutzt werden können.

3.1. Zielbestimmung

Im Folgenden werden die Ziele des zu entwickelten Systems dargestellt. Auch einige nicht notwendige Funktionen werden vorgestellt.

3.1.1. Musskriterien

Das System muss dem Nutzer eine Übersicht der gesamten Wahlergebnisse der aktuellen oder einer vorherigen Bundestagswahl in aufbereiteter Form geben. Die Wahldaten müssen dabei in einer Datenbank persistent abgelegt werden. Anhand der in dieser Datenbank abgelegten Stimmen muss das System die Sitzverteilung des Bundestags nach dem aktuellen Wahlverfahren berechnen können. Dabei müssen Sonderklauseln wie die 5-Prozent-Hürde berücksichtigt werden. Auch Überhangmandate müssen korrekt berechnet werden können. Um eine möglichst schnelle Anzeige der Wahlergebnisse zu ermöglichen sollen Wahlergebnisse in der Datenbank voraggregiert werden können. Die Ergebnisse einer Bundestagswahl müssen auch auf Wahlkreisebene angezeigt werden können. Auf Wahlkreisebene muss angezeigt werden, wie viele Stimmen die einzelnen Landeslisten der Parteien erhalten haben. Zudem muss die Stimmverteilung unter den Direktkandidaten angezeigt werden.

3.1.2. Sollkriterien

Das System soll dem Nutzer die Möglichkeit bieten, elektronisch eine Stimme abzugeben. Diese Stimmabgabe soll in einem Wahllokal mit einem dort ausgegebenen Code erfolgen. Es sollen nur Stimmen für Direktkandidaten oder Landeslisten abgegeben werden können, die in dem Wahlkreis, in dem die Stimme abgegeben wird, auch antreten. Auch die Abgabe von ungültigen Stimmen soll möglich sein. Aus Datenschutzgründen sollen abgegebene Stimmen nur anonymisiert gespeichert werden. Eingetragene Stimmen sollen auch in der Datenbank abgelegt werden und auch nach Neustart des Systems noch zur

Verfügung stehen. Sie sollen auch bei der Berechnung des Wahlergebnisses berücksichtigt werden.

3.1.3. Kannkriterien

Das System soll dem Nutzer die Möglichkeit bieten, sich die Wahlbeteiligung einer Bundestagswahl in einem Wahlkreis anzeigen zu lassen. Hier soll das System die Anzahl der Wähler im Vergleich zur Anzahl der Wahlberechtigten in dem Wahlkreis anzeigen können.

3.1.4. Abgrenzungskriterien

Das System muss die Ergebnisse von Bundestagswahlen vor 2013 nicht anzeigen können. Zudem müssen Wahlergebnisse nur nach dem aktuellen Verfahren berechnet werden können. Eine Berechnung nach anderen Verfahren, etwa zum Vergleich der Verteilung der Überhangmandate bei verschiedenen Verfahren, wird nicht benötigt. Das System muss keine Ergebnisse für die Bundestagswahl 2013 anzeigen können, die sich auf Kandidatendaten beziehen. Zu diesen Ergebnissen gehört etwa die prozentuale Anzahl der Frauen im Bundestag.

3.2. Einsatz

Das System ist für den Einsatz als Informationssystem gedacht. Über eine Webseite soll sich der Nutzer über die Ergebnisse der aktuellen oder einer vorherigen Bundestagswahl informieren können. Zudem soll das System für die elektronische Stimmabgabe eingesetzt werden können, indem es einem Nutzer die Möglichkeit bietet, eine Stimme mit einem in einem Wahllokal ausgeteilten Code abzugeben.

3.2.1. Anwendungsbereiche

Angewendet wird das System von Nutzern, die sich über die Ergebnisse einer Bundestagswahl informieren wollen. Zudem soll das System in Wahllokalen für die elektronische Abgabe von Einzelstimmen genutzt werden können.

3.2.2. Zielgruppen

Die Zielgruppe umfasst Bürger, die sich für das Ergebnis einer Bundestagswahl interessieren. Die Zielgruppe für die elektronische Abgabe von Einzelstimmen sind Bürger, die ihre Stimme in einem Wahllokal elektronisch abgeben wollen.

3.3. Umgebung

Die Software wird in der Programmiersprache Java verfasst. Sie nutzt zur Oberflächengenerierung GWT und arbeitet mit einer PostgreSQL Datenbank.

3.3.1. Software

Zum Betrieb des Servers muss mindestens Docker installiert sein. Das gleiche gilt für den Betrieb der Datenbank. Um die Anwendung benutzen zu können ist Webbrowser nötig. Unterstützt wird der Google Chrome Browser.

3.3.2. Hardware

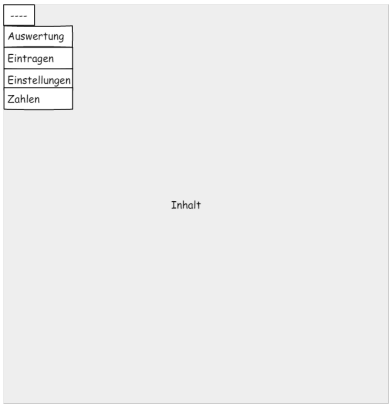
Die Systemhardware muss der Benutzerzahl angepasst werden. Die Funktionalität des Systems für geringe Benutzerzahlen kann mit 8 GB freiem RAM, 30 GB freiem Festplattenspeicher und einer Intel CPU vom Typ Core i5-6000 oder besser gewährleistet werden.

3.4. Funktionalität

Im folgenden werden einzelne Funktionen des Systems näher vorgestellt.

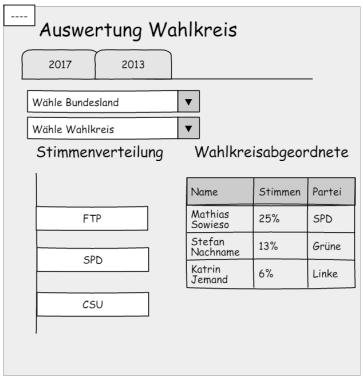
3.4.1. Anzeige des Wahlergebnisses einer Bundestagswahl

- Primärer Akteur: Bürger
- Trigger: Bürger möchte Informationen über Ergebnisse einer Bundestagswahl
- Haupterfolgsszenario:
    1. Nutzer wählt die Option ”Parlament”
    2. System berechnet anhand von voraggregierten Wahlergebnissen Ergebnis der Bundestagswahl
    3. System zeigt Wahlergebnis bestehend aus prozentualer und absoluter Verteilung von Stimmen auf Parteien, Sitzverteilung im Bundestag, Anzahl der Überhangmandate und Mitglieder des Bundestages an
- Alternative Szenarien:
    1. a) Nutzer möchte sich die Ergebnisse einer vorherigen Bundestagswahl anzeigen lassen und wählt deshalb statt dem voreingestellten Jahr der letzten Bundestagswahl ein vorheriges
- Oberfläche Beispiel:



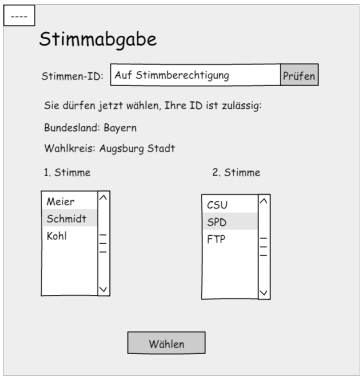
3.4.2. Anzeige des Wahlergebnisses einer Bundestagswahl in einem Wahlkreis

- Primärer Akteur: Bürger
- Trigger: Nutzer möchte sich über die Ergebnisse einer Bundestagswahl in einem Wahlkreis informieren
- Haupterfolgsszenario:
  1. Nutzer wählt die Option ”District”
  2. Nutzer wählt Wahlkreis aus einer Vorschlagsliste
  3. System zeigt Wahlergebnis des Wahlkreises bestehend aus prozentualer und absoluter Verteilung von Stimmen auf Landeslisten und Direktkandidaten und die Anzahl der Wähler und Nichtwähler. Diese Daten werden im Vergleich zu den Daten der vorherigen Bundestagswahl dargestellt.
- Oberfläche Beispiel:



3.4.3. Abgabe einer Einzelstimme

- Primärer Akteur: Wähler
- Trigger: Nutzer möchte bei einer Bundestagswahl seine Stimme in elektronischer Form abgeben
- Vorbedingung: Das System zeigt ein Eingabefeld für einen Autorisierungscode für die elektronische Stimmabgabe an
- Haupterfolgsszenario:
  1. Nutzer gibt Authentifizierungscode an
  2. System prüft Code
  3. System zeigt Stimmzettel mit wählbaren Direktkandidaten und Landeslisten an
  4. Nutzer vergibt Erststimme an einen Direktkandidaten
  5. Nutzer vergibt Zweitstimme an eine Landesliste
- Alternative Szenarien:
  - Nutzer möchte Erst- oder Zweitstimme nicht an einen Kandidaten oder eine Landesliste vergeben
    - Nutzer wählt keinen Kandidaten oder keine Landesliste
  - Authentifizierungscode ist ungültig
    - System informiert Nutzer entsprechend und fordert ihn zur erneuten Eingabe des Codes auf
- Oberfläche Beispiel:



i

3.4.4. Nichtfunktionale Anforderungen

Im Folgenden sind die nichtfunktionalen Anforderungen spezifiziert:

3.4.5. Sicherheit

Um die Sicherheit zu gewährleisten muss der Betreiber des Systems folgendes beachten:

- Die Datenbank darf in Produktivsystemen nur aus einem abgeschotteten Netz zugreifbar sein. In diesem darf sich außer der Datenbank nur der Server befinden.
  - Wenn nötig, darf ein autorisierter Techniker Wartungsarbeiten im Netz vornehmen, allerdings muss der Betreiber geeignete Maßnahmen ergreifen, um Datenintegrität und den Datenschutz gegenüber diesem sicherzustellen.
- Die Eingabe sensibler Daten darf nur in einer gesicherten Umgebung möglich sein. Verschiedene Möglichkeiten zur Umsetzung durch den Betreiber sind denkbar:
  - Verwenden in einem abgeschotteten Netz
  - Sicherung mittels aktueller Verschlüsselungstechnik und PFS[1], zum Beispiel mittels TLS[2]

Datenschutz

Der Datenschutz ist gewährleistet, da die Anwendung sensible Daten (insbesondere einzelne Stimmen) nur in akkumulierter Form an den Nutzer weiter gibt.

Manipulationssicherheit

Die Daten können nur durch Stimmenabgabe und direkte Manipulation an der Datenbank manipuliert werden. Letzteres muss vom Betreiber ausgeschlossen werden. Um ersteres zu gewährleisten, muss der Betreiber zur Stimmenabgabe einen unkompromittierten Kommunikationsweg sicherstellen.

Die Abgabe einer einzelnen Stimme direkt im Wahlbüro wird mit einem Sicherheitscode autorisiert, sodass nur autorisierte Personen die Stimmenabgabe durchführen können. Ein Sicherheitscode berechtigt *einmalig* zur Stimmenabgabe. Ein Sicherheitscode ist entweder für eine Einzelstimme oder für die akkumulierte Stimmabgabe gültig, nie jedoch für beides. Ein Sicherheitscode ist ausreichend lang, um Brute-Force Attacken auf aktueller Hardware (Stand Oktober 2017) zu verhindern.

3.4.6. Robustheit und Verfügbarkeit

Das System garantiert eine Uptime von 90%, akkumuliert auf ein Jahr, sofern die Systemumgebung zur Verfügung steht. Bei illegalen Eingaben in der Benutzerschnittstelle garantiert das System eine Robustheit.

[1] Perfect Forward Secrecy
[2] Transport Layer Security

3.4.7. Korrektheit

Die Analysen des Systems sind nach abgeschlossener Wahl bis auf die dritte signifikante Stelle korrekt. Sollte die dritte signifikante Stelle einer Zahl von weniger als 10 Einzelstimmen abhängen, kann die Korrektheit der Analyse nicht garantiert werden.

3.4.8. Nutzbarkeit

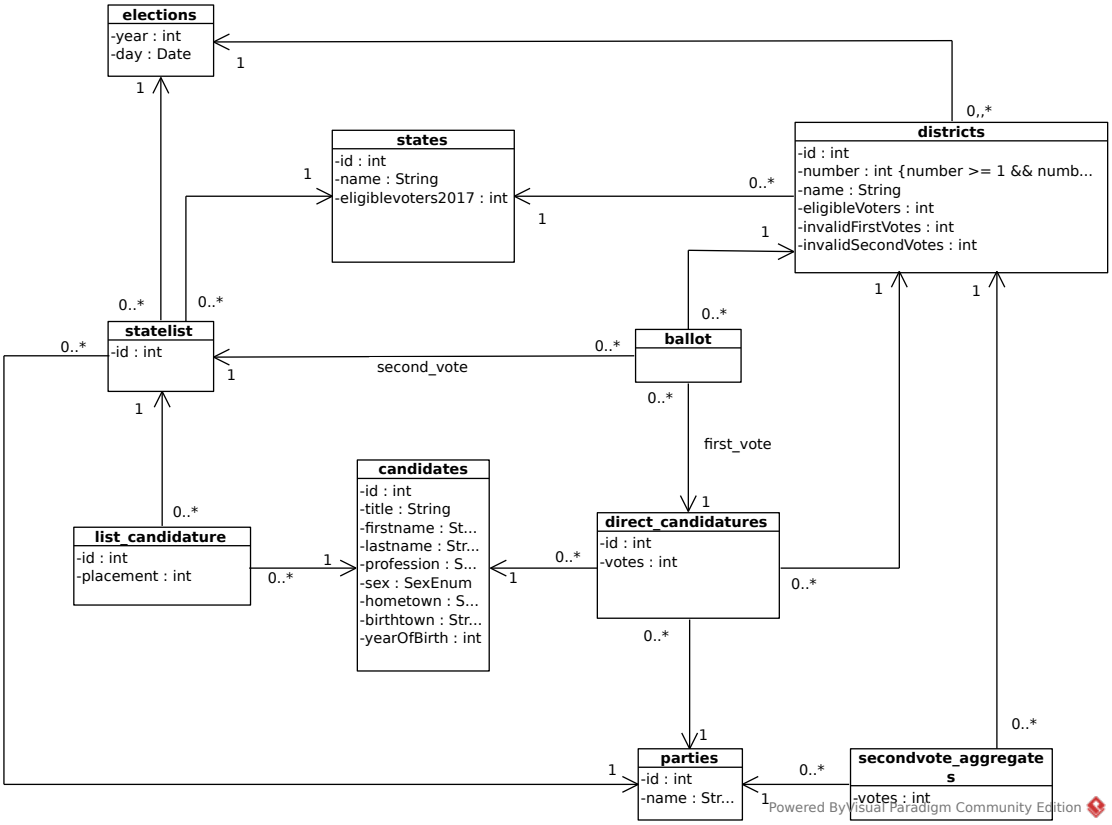
Der Nutzer muss die Applikation ohne Zuhilfenahme einer Dokumentation verstehen. Auch ohne technische Vorkenntnisse soll er nach 15 Minuten fähig sein mit der Software umzugehen. Da die elektronische Abgabe von Einzelstimmen in Wahllokalen erfolgt soll hierfür keine Einarbeitungszeit nötig sein. Der Nutzer sollte für Authentifizierung und Stimmabgabe nicht länger als 5 Minuten brauchen.

3.4.9. Datengenerator

Um die Funktionalität des Systems nachzuweisen soll ein Datengenerator implementiert werden. Dieser Generator soll die Erstellung von Stimmen, bestehend aus Erst- und Zweitstimme, ermöglichen, die den Stimmen letzten Bundestagswahl entsprechen. Aufgrund dieser Stimmen soll das korrekte Wahlergebnis berechnet werden können.

3.5. Daten

Die Daten, die das System langfristig speichern soll können dem abgebildeten Datenmodell entnommen werden.


elections
-year : int
-day : Date
states
-id : int
-name : String
-eligiblevoters2017 : int
districts
-id : int
-number : int {number >= 1 && numb...
-name : String
-eligibleVoters : int
-invalidFirstVotes : int
-invalidSecondVotes : int
statelist
-id : int
ballot
second_vote
first_vote
candidates
-id : int
-title : String
-firstname : St...
-lastname : Str...
-profession : S...
-sex : SexEnum
-hometown : S...
-birthtown : Str...
-yearOfBirth : int
list_candidature
-id : int
-placement : int
direct_candidatures
-id : int
-votes : int
parties
-id : int
-name : Str...
secondvote_aggregates
-votes : int
Powered By Visual Paradigm Community Edition

A. Benchmark-Ergebnisse

A.1. Methodik

Die folgenden Benchmarks wurden auf einem Intel i5-3570K (3,4 GHz) System mit 16 Gigabyte RAM unter Linux ausgeführt. Dabei lief jMeter zusammen mit der Anwendung sowie der Datenbank auf dem gleichen System. Von den zur Verfügung stehenden 16 Gigabyte wurden während der Tests – inklusive Betriebssystem – nur ca. 3 Gigabyte genutzt.

Jeder durchgeführte Benchmark-Lauf hat 1 Minute lang mit mehreren simulierten Clients zufällig verteilte Anfragen für die Queries Q1 bis Q6 an den Server geschickt. Jeder simulierte Client hat nach jeder Anfrage eine variable Zeitspanne gewartet, bevor er die nächste Anfrage gestartet hat. Diese variable Zeitspanne ist zufällig gewählt aus einer Basiszeitspanne und dem 1.5-fachen dieser Basiszeitspanne. Es wurden Benchmark-Läufe mit jeder Kombination aus 4, 20, 100 und 500 simulierten Clients sowie einer Basiszeitspanne von 1, 2, 4 und 8 Sekunden durchgeführt. Jedes der folgenden Diagramme zeigt Boxplots, in welchen die Antwortzeiten für eine Query und eine Basiszeitspanne nach der Anzahl der simulierten Clients dargestellt sind.

Die jMeter-Konfiguration für die Benchmarks findet sich im Repository, damit können die Benchmarks leicht auf einem beliebigen System durchgeführt werden.

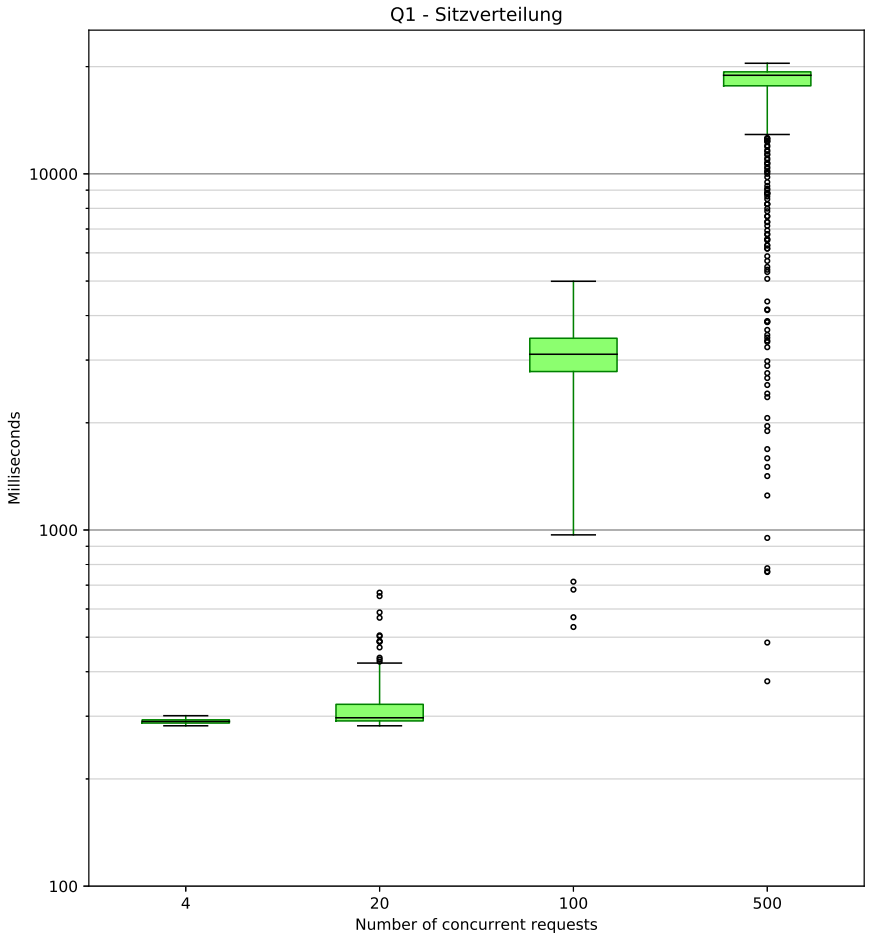


Abbildung A.1.: Q1 - Sitzverteilung (1s)

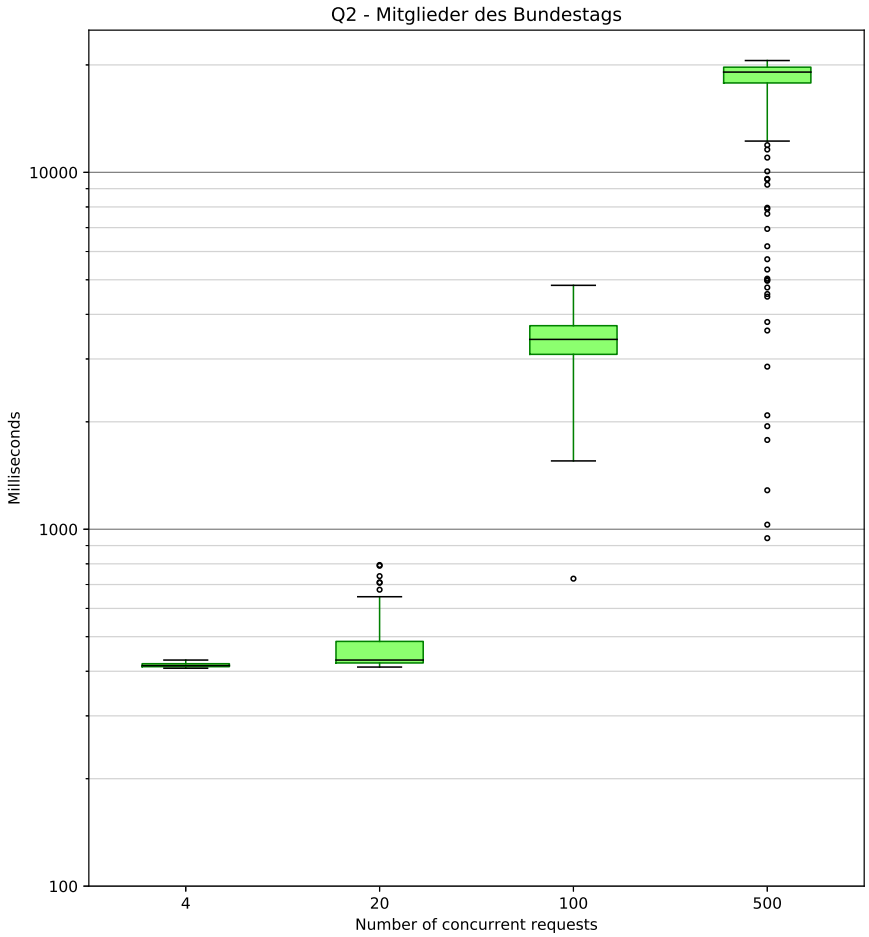


Abbildung A.2.: Q2 - Mitglieder des Bundestags (1s)

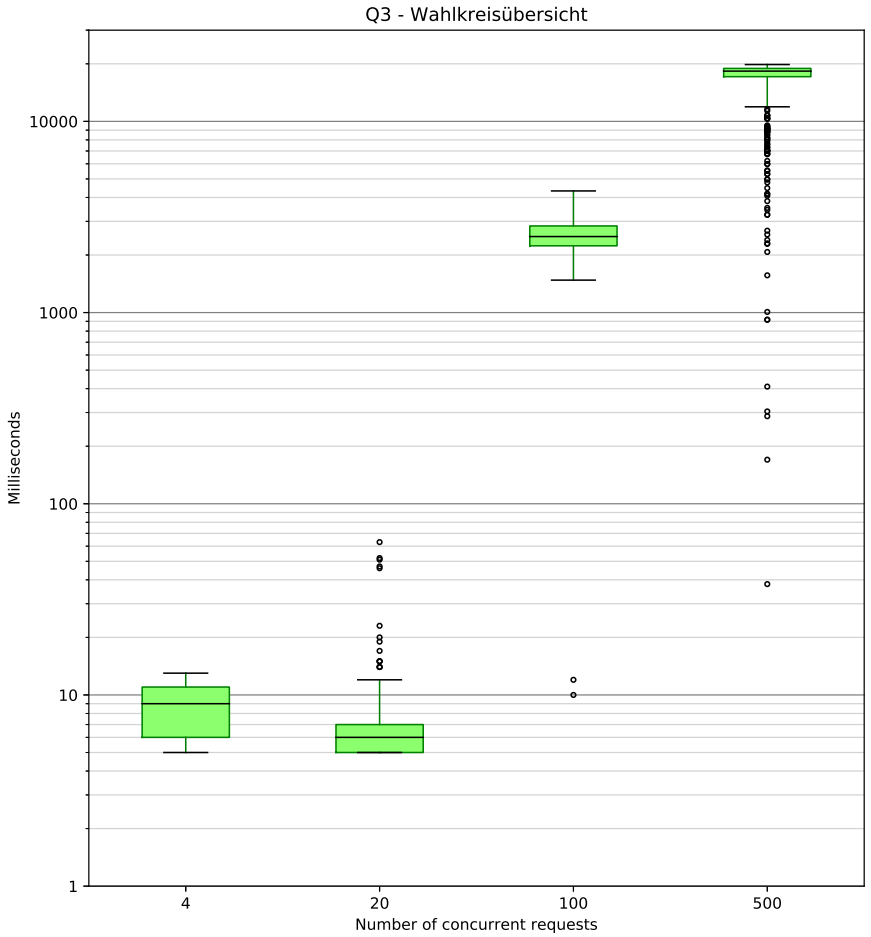


Abbildung A.3.: Q3 - Wahlkreisübersicht (1s)

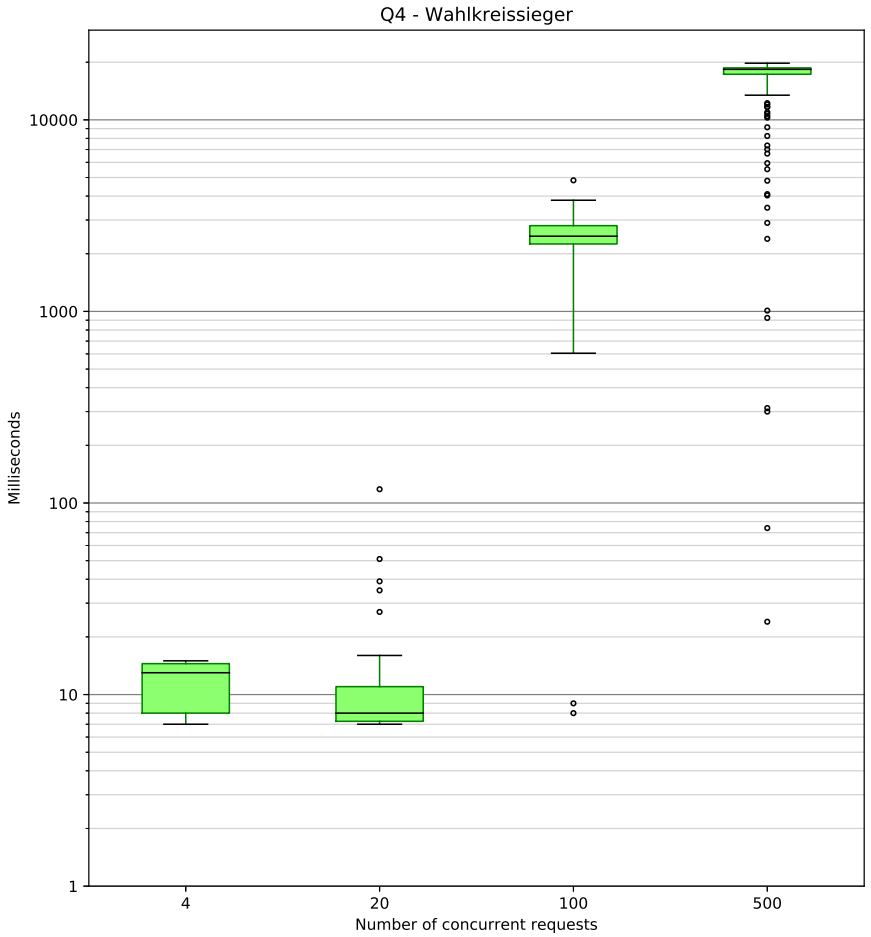


Abbildung A.4.: Q4 - Wahlkreissieger (1s)

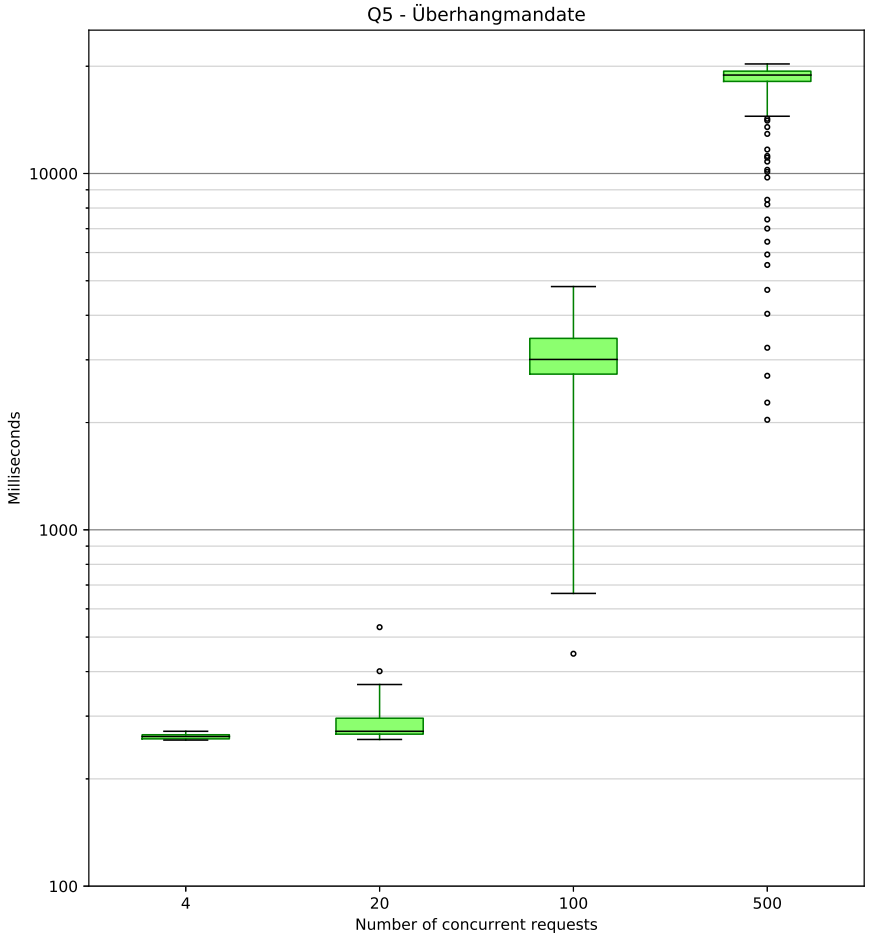


Abbildung A.5.: Q5 - Überhangmandate (1s)

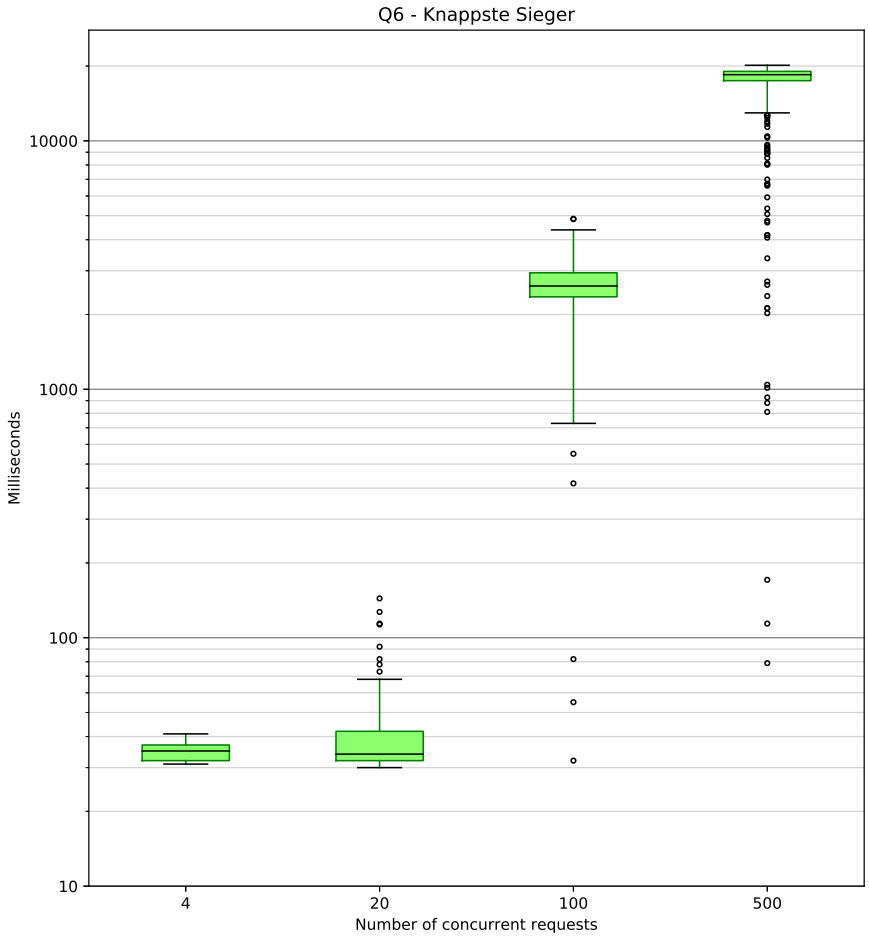


Abbildung A.6.: Q6 - Knappste Sieger (1s)

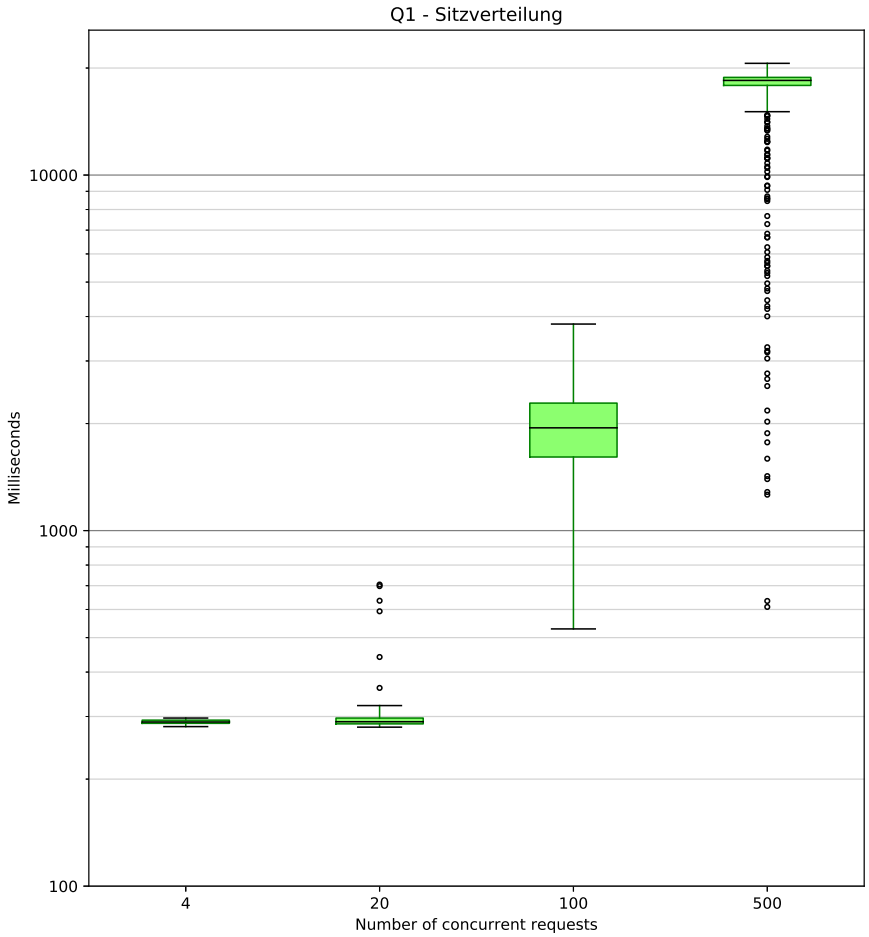


Abbildung A.7.: Q1 - Sitzverteilung (2s)

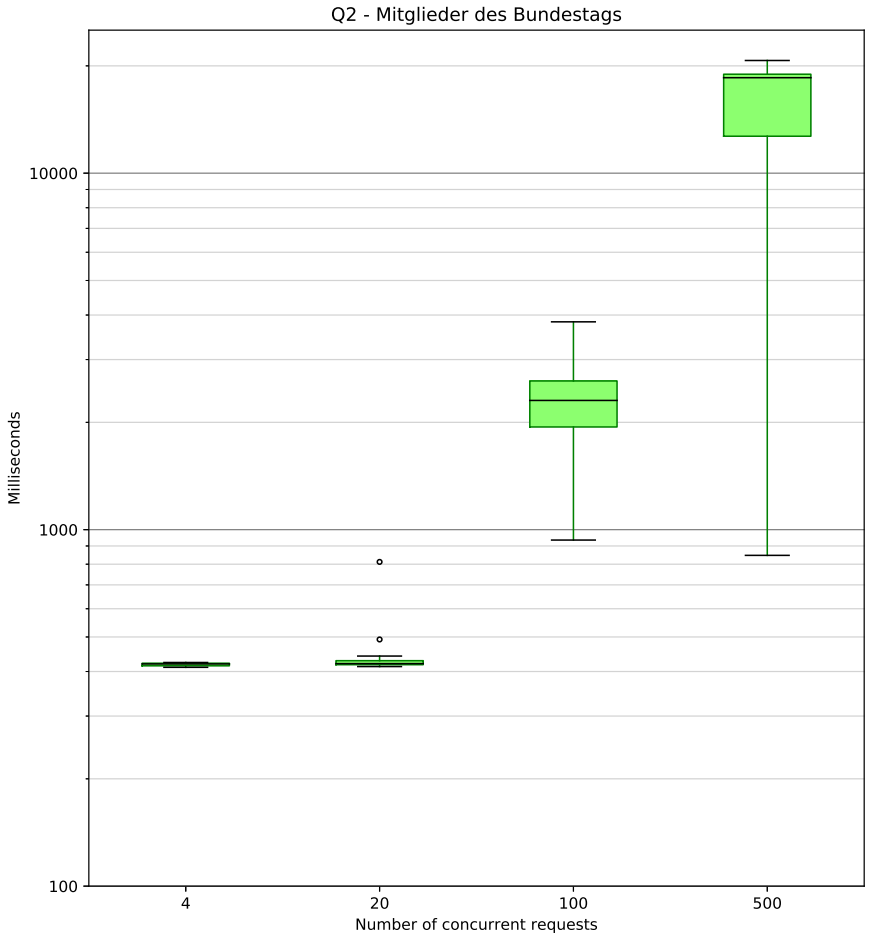


Abbildung A.8.: Q2 - Mitglieder des Bundestags (2s)

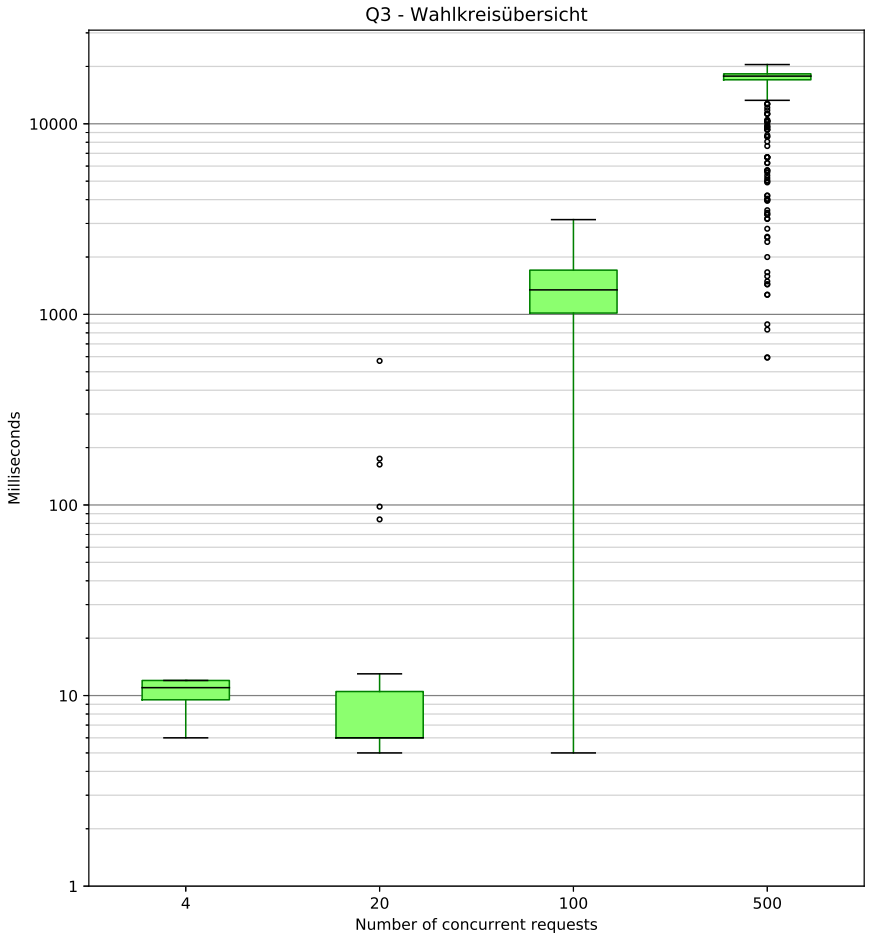


Abbildung A.9.: Q3 - Wahlkreisübersicht (2s)

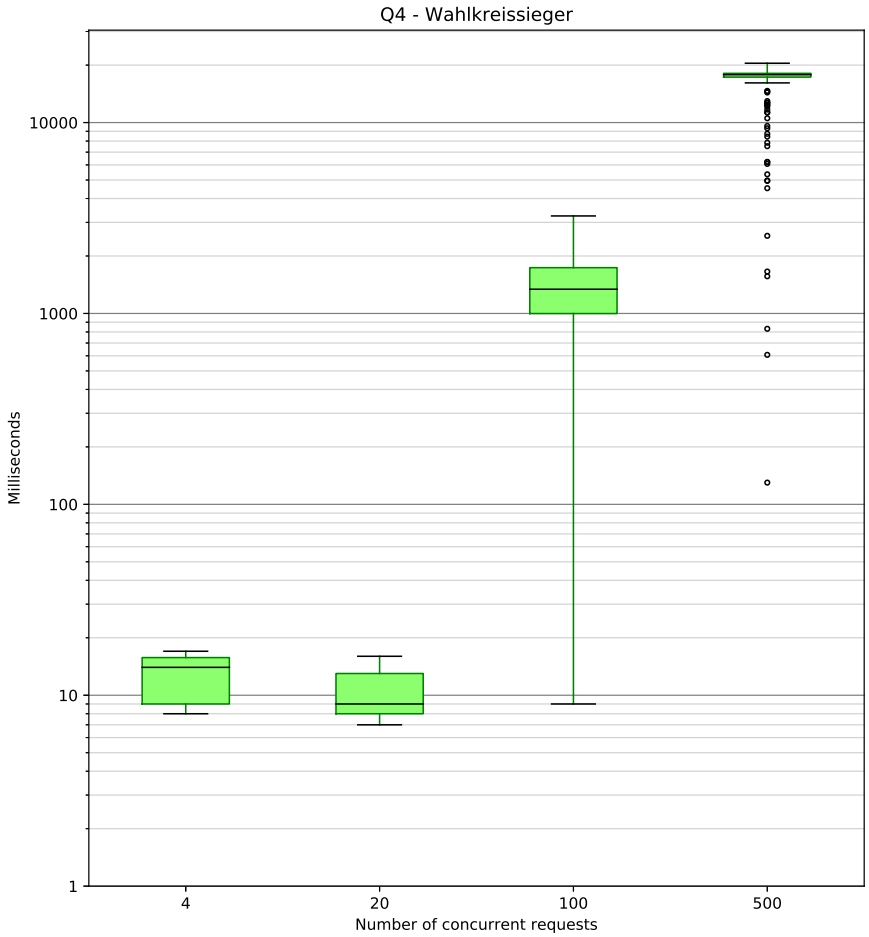


Abbildung A.10.: Q4 - Wahlkreissieger (2s)

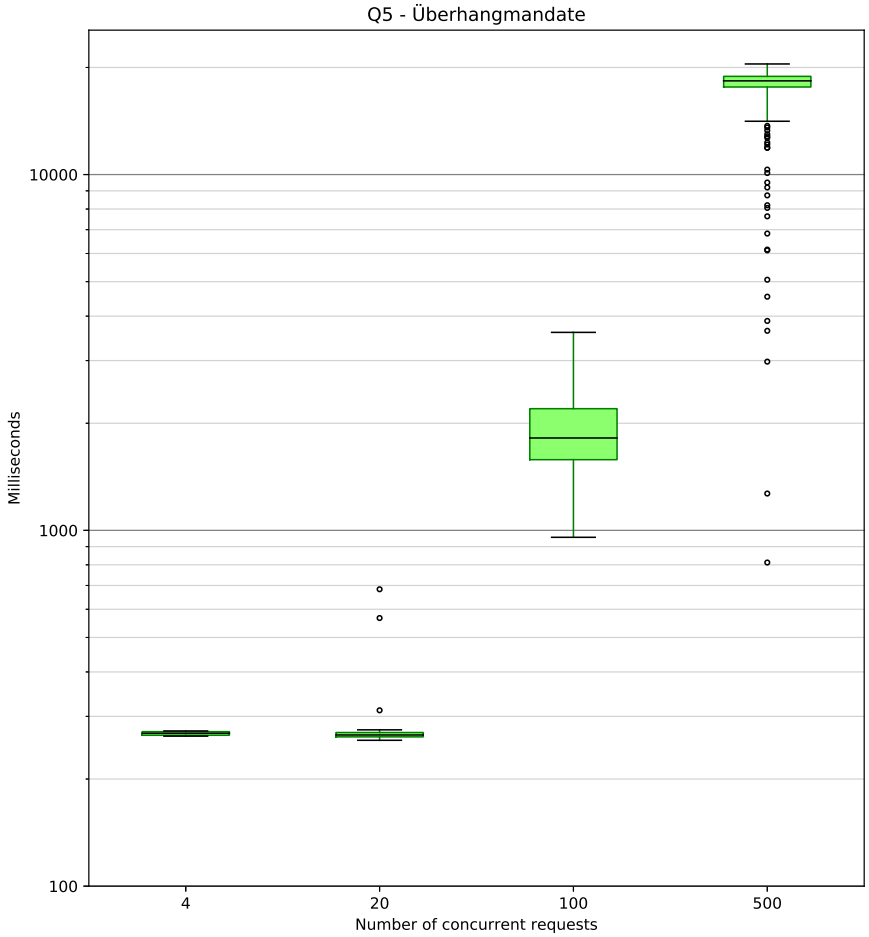


Abbildung A.11.: Q5 - Überhangmandate (2s)

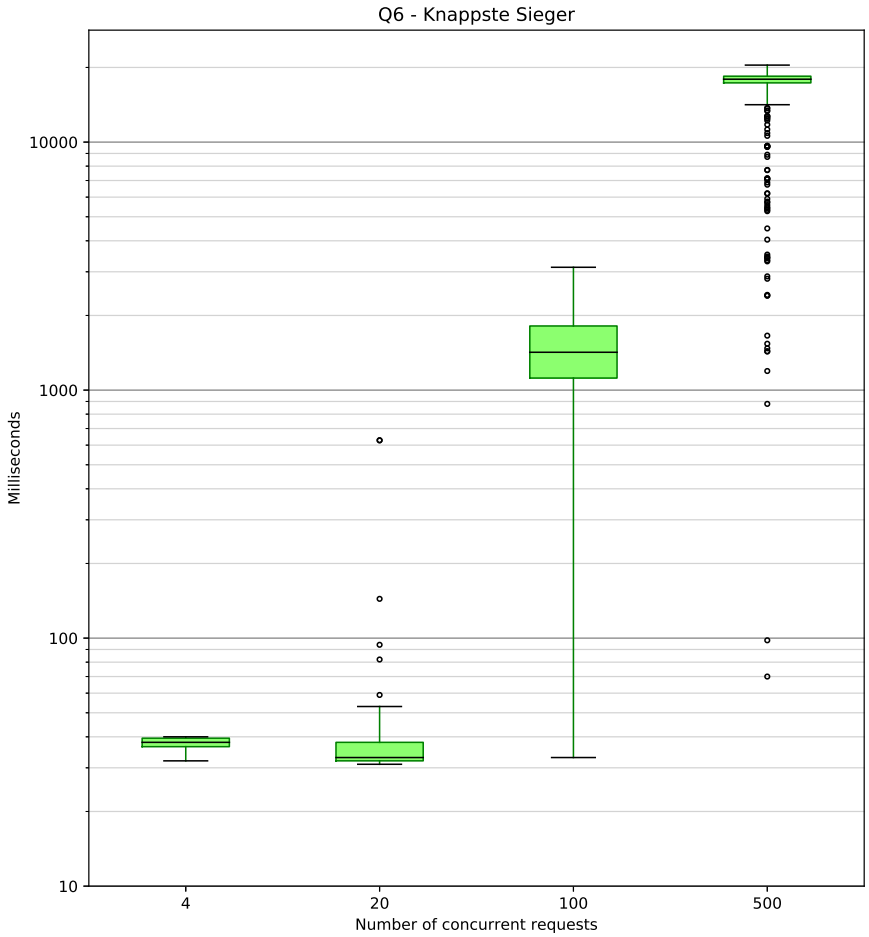


Abbildung A.12.: Q6 - Knappste Sieger (2s)

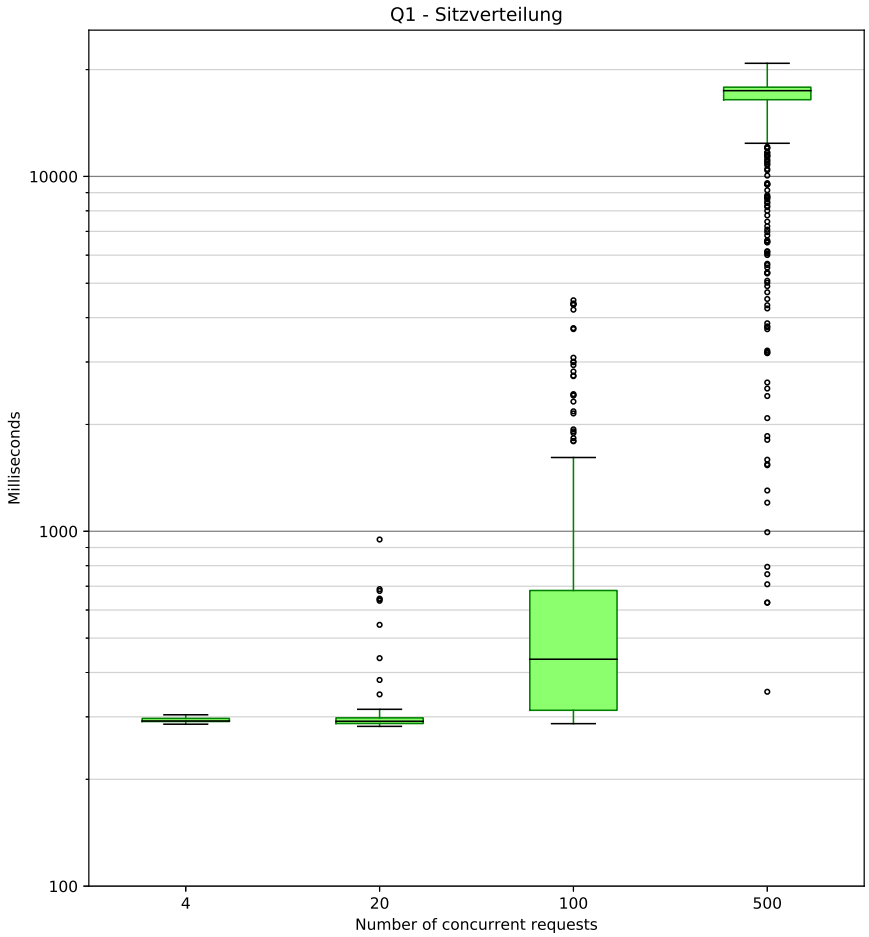


Abbildung A.13.: Q1 - Sitzverteilung (4s)

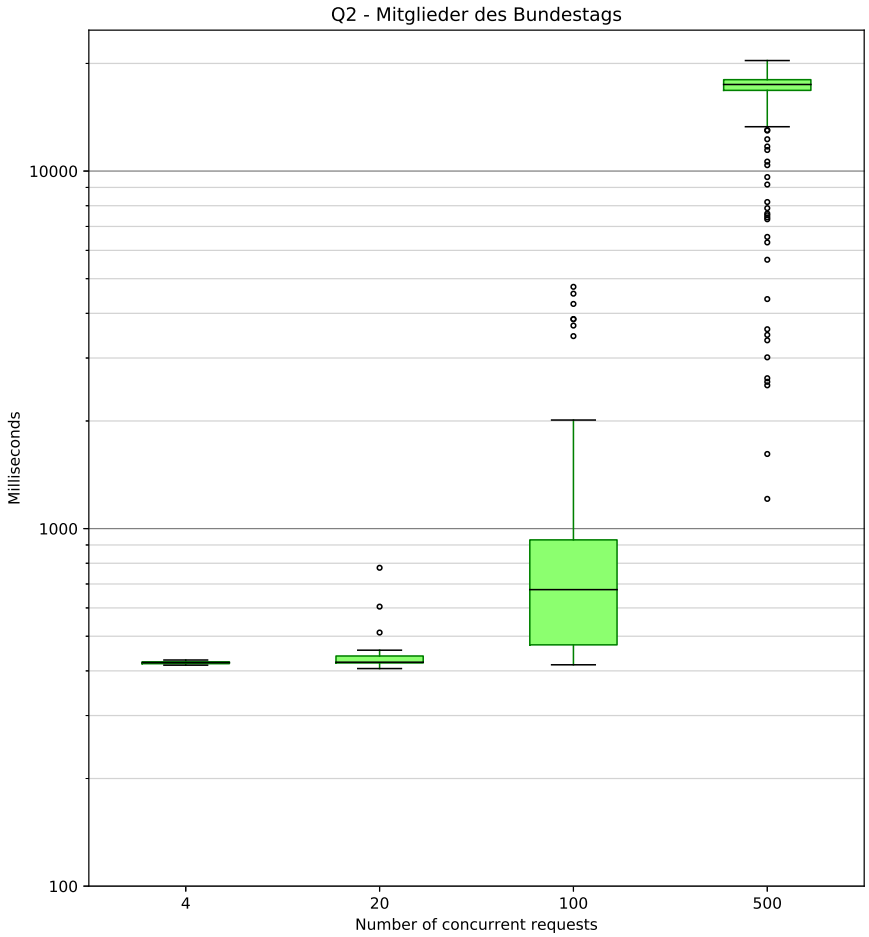


Abbildung A.14.: Q2 - Mitglieder des Bundestags (4s)

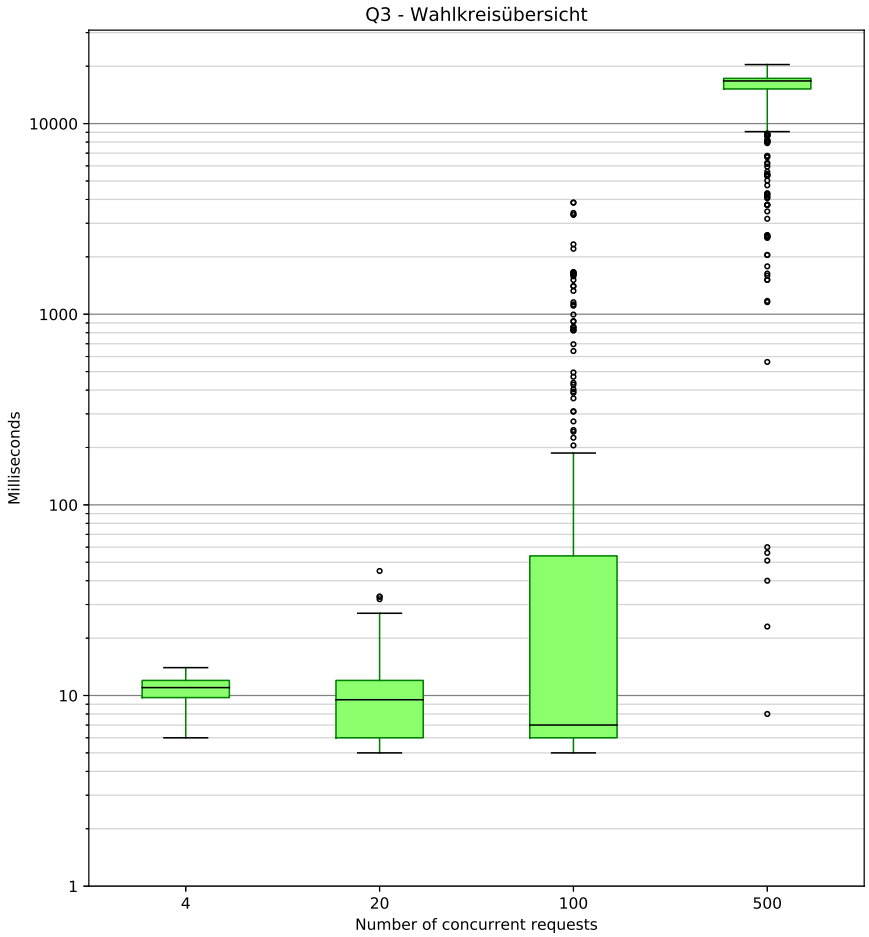


Abbildung A.15.: Q3 - Wahlkreisübersicht (4s)

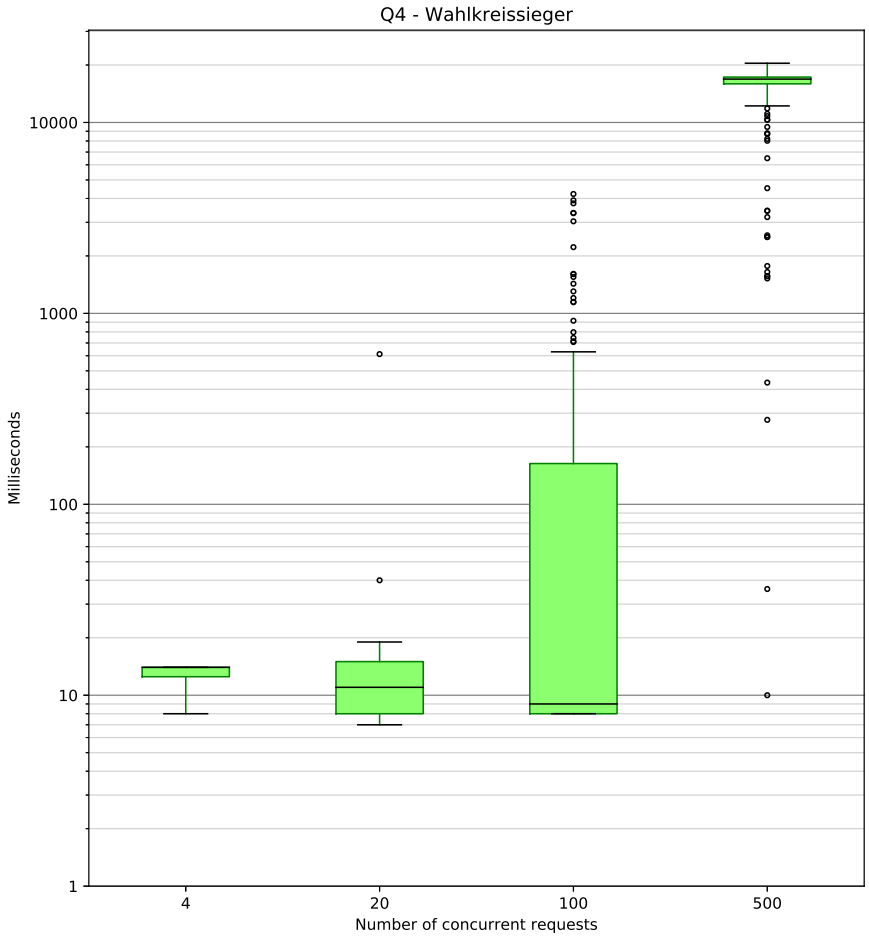


Abbildung A.16.: Q4 - Wahlkreissieger (4s)

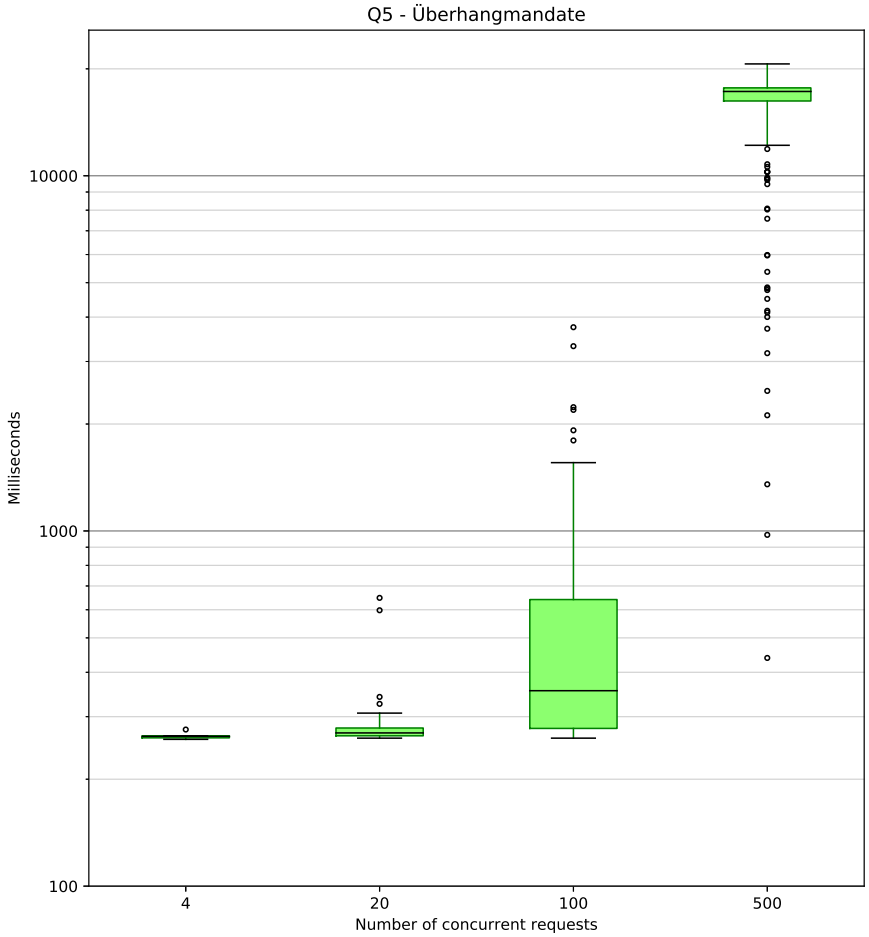


Abbildung A.17.: Q5 - Überhangmandate (4s)

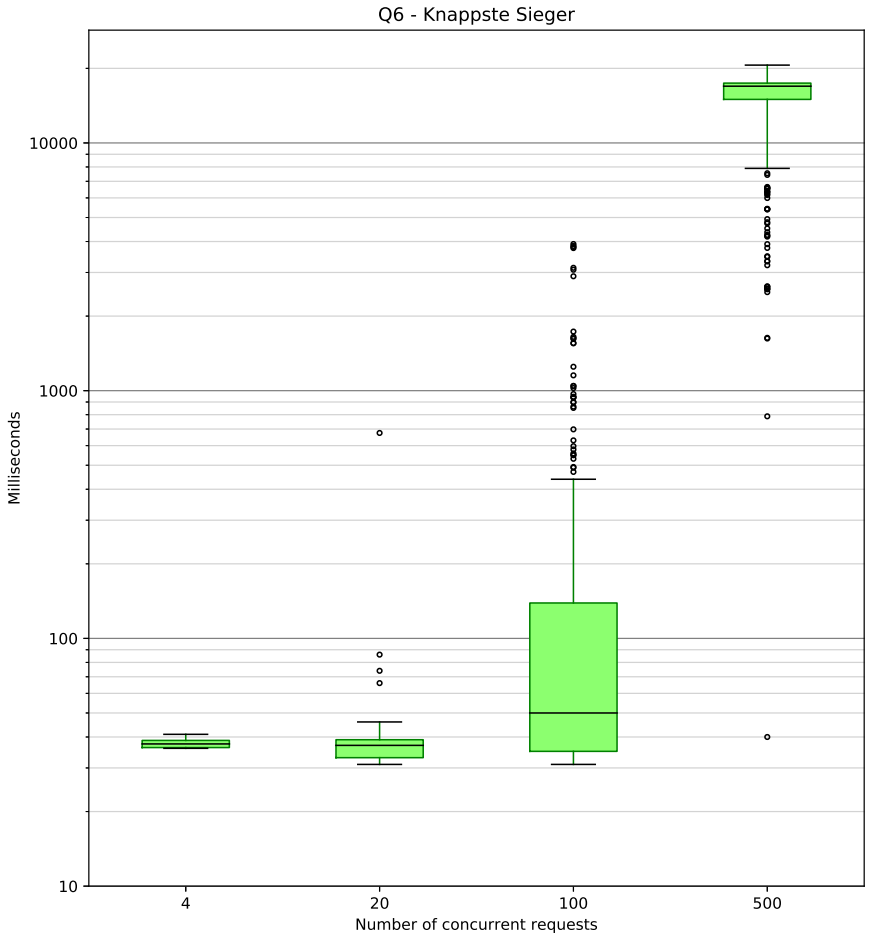


Abbildung A.18.: Q6 - Knappste Sieger (4s)

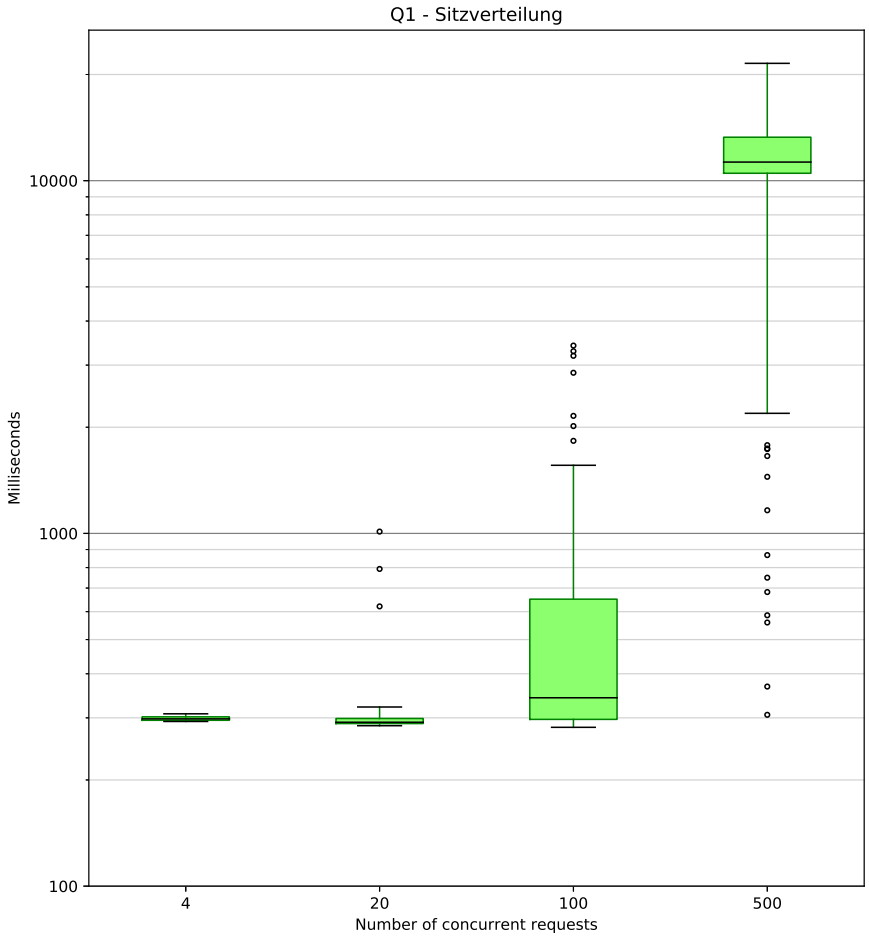


Abbildung A.19.: Q1 - Sitzverteilung (8s)

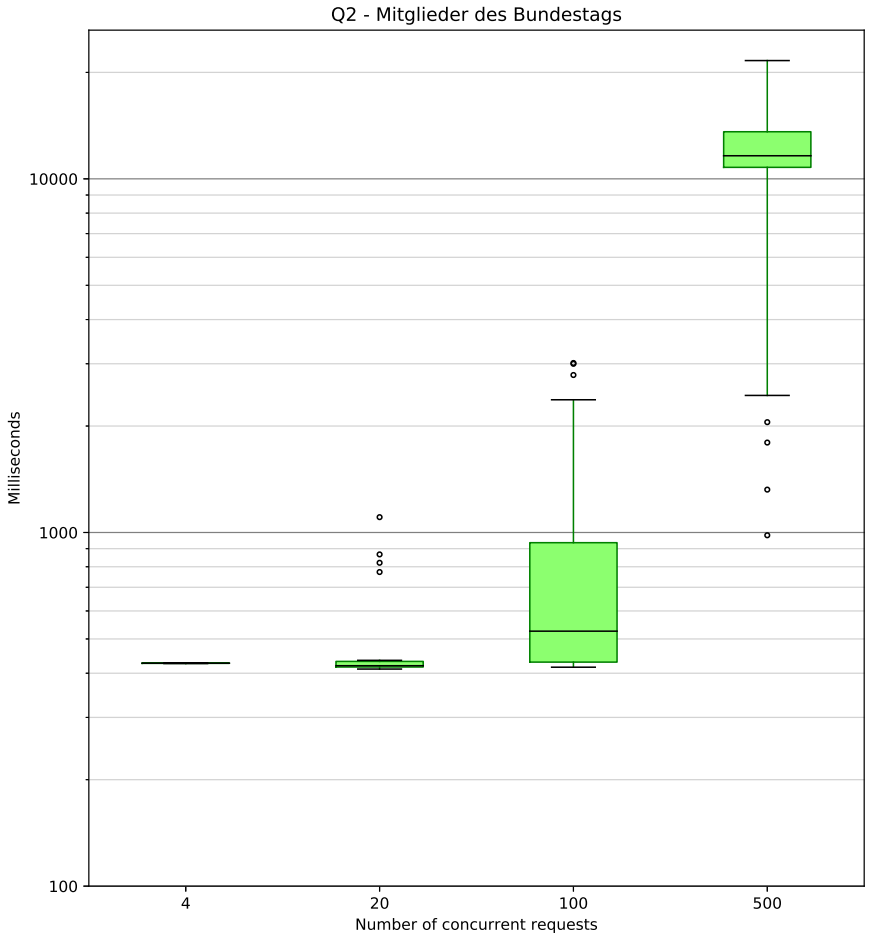


Abbildung A.20.: Q2 - Mitglieder des Bundestags (8s)

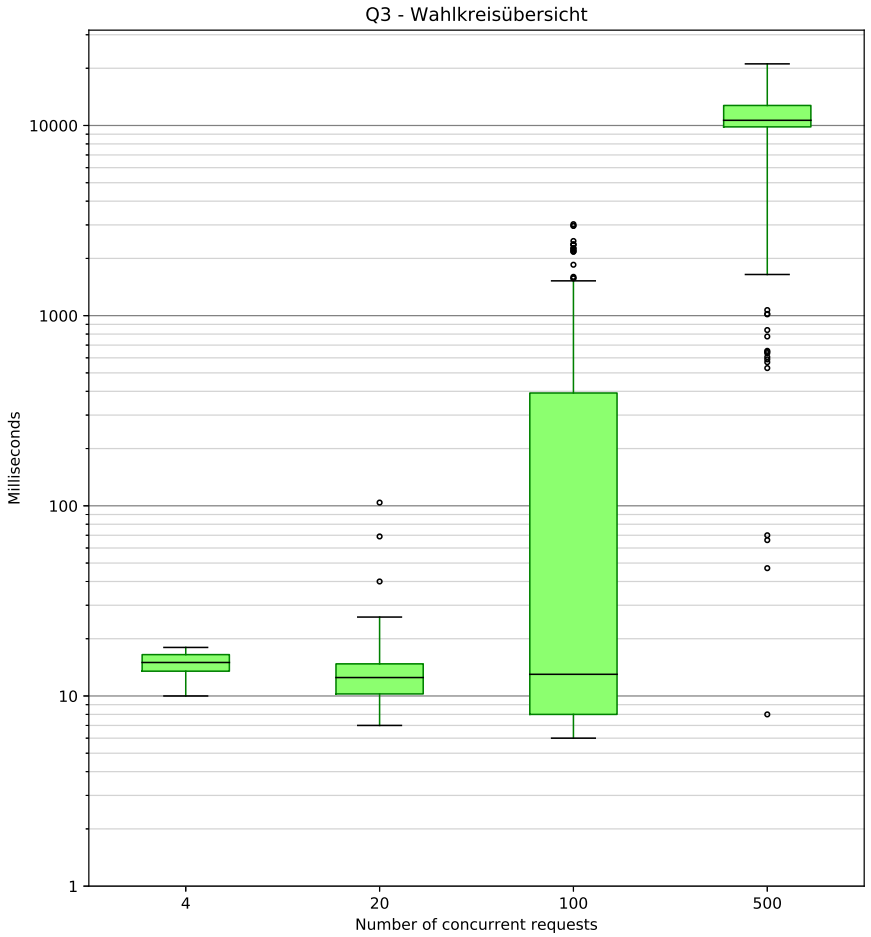


Abbildung A.21.: Q3 - Wahlkreisübersicht (8s)

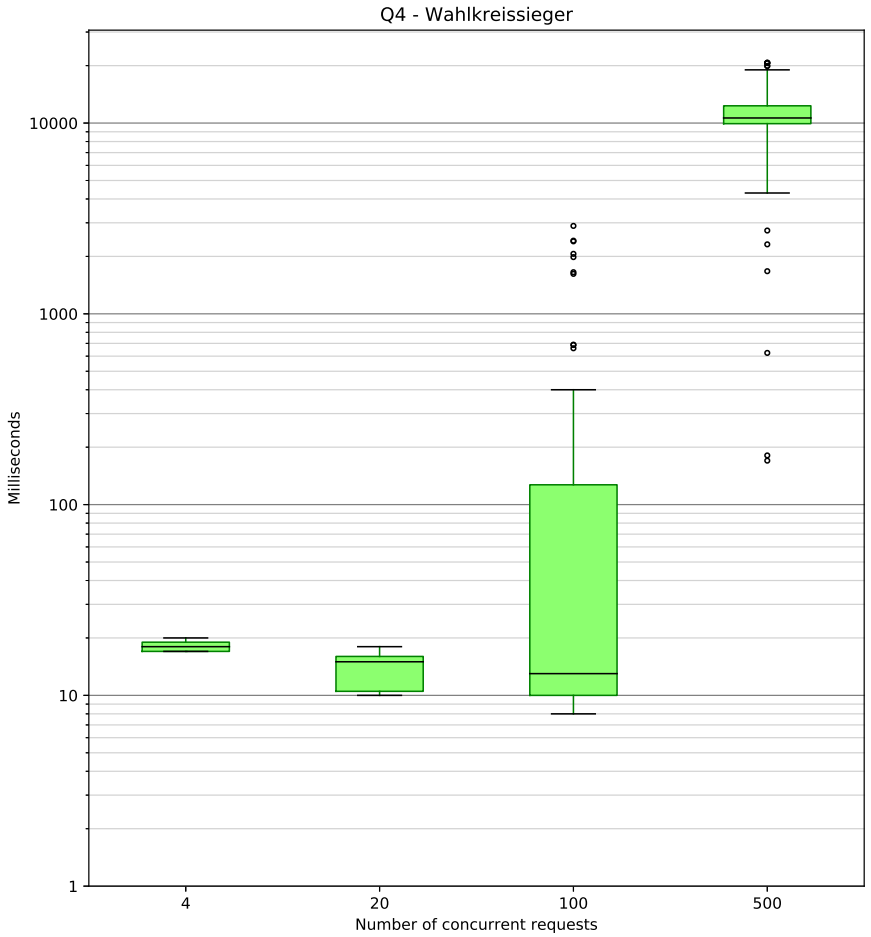


Abbildung A.22.: Q4 - Wahlkreissieger (8s)

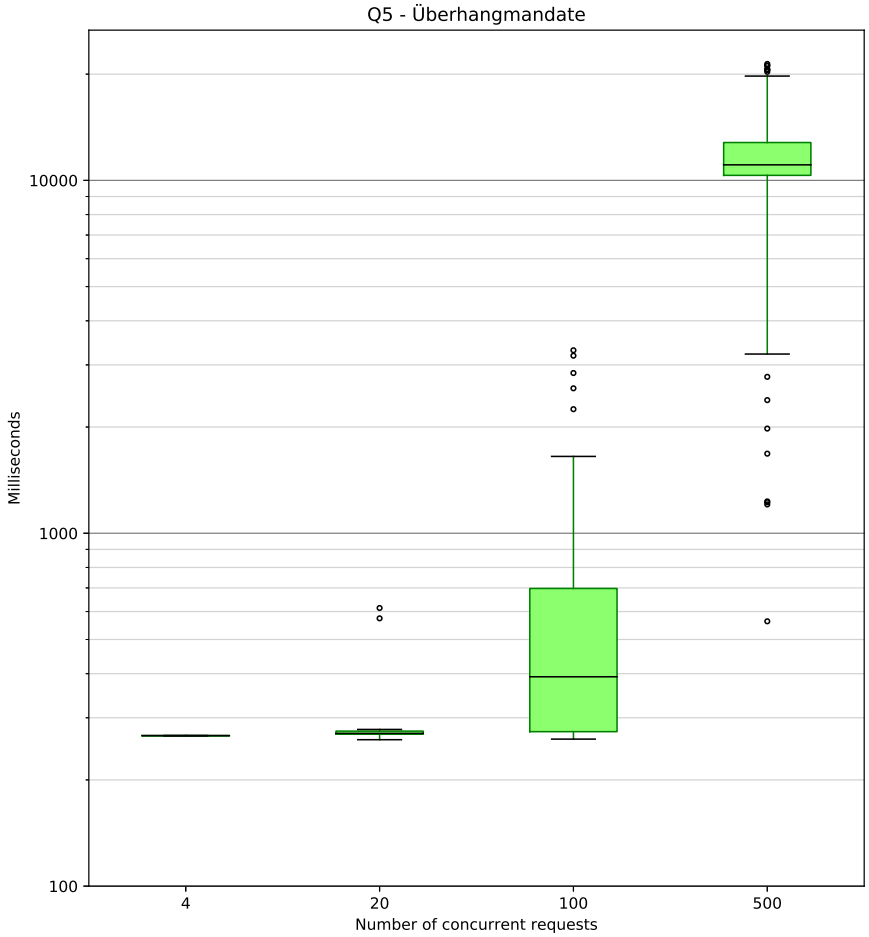


Abbildung A.23.: Q5 - Überhangmandate (8s)

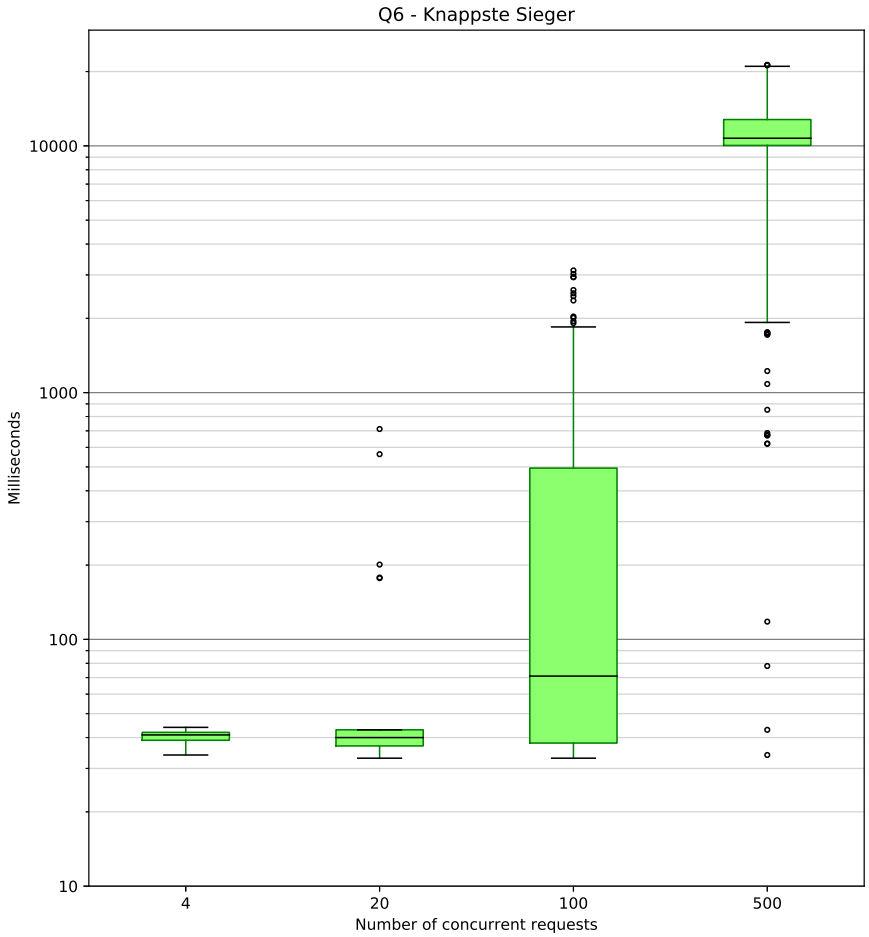


Abbildung A.24.: Q6 - Knappste Sieger (8s)