

# Deckblatt folgt noch

Version 1.1

## Inhaltsverzeichnis

---

Abbildungsverzeichnis.....	IV
Tabellenverzeichnis.....	IV
Abkürzungsverzeichnis.....	V
1. Einleitung.....	1
2. Google Fusion Tables.....	2
2.1. Überblick.....	2
2.2. Systemarchitektur.....	3
2.2.1. Speicher-Stack.....	3
2.2.2. Zeilenspeicher.....	4
2.2.3. Schema-Speicher.....	4
2.2.4. View-Speicher.....	4
2.2.5. Kommentar-Speicher.....	4
2.3. Datenabfrage.....	5
2.4. Transaktionen.....	5
2.5. Datenvisualisierung.....	5
2.5.1. Visualisierungsinfrastruktur.....	5
2.5.2. Einbettung.....	5
2.6. Geografische Features.....	5
2.6.1. Google Maps-Infrastruktur.....	5
2.6.2. Einbettung.....	5
2.7. Google Fusion Tables API.....	5
3. Projektvorstellung.....	6
3.1. Anforderungen.....	6
3.1.1. Suche nach Weihnachtsmärkten.....	6
3.1.2. Einreichung weitere Weihnachtsmärkte.....	8
3.1.3. Infrastruktur.....	9
3.2. Ziele.....	10
4. Literatur/Quellen.....	11
4.1. Conference Papers & Publications.....	11
4.2. Presentations & workshops by the Fusion Tables team.....	12

# Änderungsprotokoll

[illegible]

## **Abbildungsverzeichnis**

---

## **Tabellenverzeichnis**

---

## Abkürzungsverzeichnis

---

Abkürzung	Bedeutung / Beschreibung
SOAP	Simple Object Access Protocol
REST	Representational State Transfer

## 1. Einleitung

---

Manuel und ich machen ein Projekt mit Google Fusion Tables und befinden uns mitten in der Einarbeitung und dem Aufsetzen der Struktur. Ziel ist es eine Art Datencockpot für Berliner Weihnachtsmärkte zu entwickeln, in dem je nach Bezirk verschiedene Weihnachtsmärkte gefiltert werden können. Darüber hinaus auch zusätzliche Eigenschaften wie „sind Fahrgeschäfte vorhanden“ oder „Glühweinpreise“. Unsere Projektplanung und die Meilensteine sind ebenfalls definiert. Nach der Einarbeitung folgt die Phase der Spezifikation mit und anschließend die Vorbereitung und beschaffen der benötigten Geo-Koordination für unser Projekt. Anschließend folgt die Umsetzungsphase wo die Google Fusion Tables mit Google Maps kombiniert und in ein HTML-Prototypen integriert werden. Um die Google Fusion Tables auch per Webinterface zu befüllen arbeiten wir zusätzlich mit dem Open Data Kit. Nach dem alles umgesetzt wurde folgt die abschließende Dokumentation und Vorbereitung für die Präsentation. (Status - 15. November)

## 2. Google Fusion Tables

---

### 2.1. Überblick

Google Fusion Tables ist ein 2009 gestarteter cloud-basierter Webdienst zur Verwaltung, Integration und insbesondere zur Visualisierung tabellenbasierter Daten. Der Service unterstützt Unternehmen bei der internen wie externen Bereitstellung von Daten und zielt darauf ab, die Zusammenarbeit von Anwendern aus verschiedenen Organisationseinheiten zu vereinfachen. Für die grafische Aufbereitung der Werte ist dabei neben herkömmlichen Diagrammen auch die geospatiale Projektion der Daten in das Kartenmaterial der Geodatendienste Google Maps und Google Earth möglich.

Google Fusion Tables erlaubt den Upload verschiedener Datenformate (csv, kml, etc.) über den Service Google Drive (bis zu 250 MB). Für eine möglichst benutzerfreundliche Auswertung hält das Framework Aggregationsfunktionalitäten bereit und schlägt auf Basis der hinterlegten Daten automatisch passende Visualisierungsformen vor. User können ihre Daten für die Zusammenarbeit mit anderen Anwendern (und für Suchmaschinen) freigeben und dank einer Diskussionsfunktion sogar bis auf Zeilen-, Spalten- und Zellenebene detailliert kommentieren. Neben der Integration verschiedener Datenquellen können so zu Auswertungszwecken auch Datentabellen verschiedener Nutzer verbunden werden.

Die Entwicklung von Google Fusion Tables folgt letztlich folgenden drei Leitlinien:

1. Vereinfachung der Datenverwaltung für Anwender ohne Erfahrung im Umgang mit Datenbanksystemen
2. Erhöhung der Attraktivität von Data-Sharing und Informationsintegration durch aussagekräftige Visualisierungen
3. Unterstützung von cloud-basierter Kollaboration, die über reine Datenabfrage hinausgeht (z.B. durch Diskussions- und Abstimmungsfeatures)

## **2.2. Systemarchitektur**

Die Systemarchitektur von Google Fusion Tables wird in drei Schichten eingeteilt: Frontend Dispatcher, Query Processing Module und Backend. Anfragen (Requents) können grundsätzlich von verschiedenen Clients gestellt werden (Google Fusion Tables-Website, Stand-Alone-Anwendungen, eingebettete Visualisierungen, etc.). Der Frontend Dispatcher übersetzt die Anfragen und leitet die standardisierten Queries an das Query Processing Module weiter. Das Processing Module erstellt einen Query Plan und leitet ihn ans Backend weiter. Im Backend werden die Queries abgearbeitet und die Ergebnisse an das Query Processing Module und schlussendlich an das Frontend zurückgeliefert. Als Persistenzschicht dienen dabei synchron replizierte Bigtable-Server. Die größte Herausforderung der Datenbankschicht besteht in der Bearbeitung hunderttausender Tabellen mit unterschiedlichen Schemata, Größen und Anfragelast-Statistiken.

### **2.2.1. Speicher-Stack**

Google Fusion Tables basiert auf zwei Ebenen des Google Speicher Stacks: Bigtable und Megastore.

Bigtable ist Googles proprietäres Hochleistungs-DBMS und speichert in erster Linie Tabellen als komplexe und nicht zwingend normalisierte (key, value)-Tupel. Diese werden nach Schlüsselwerten sortiert über mehrere Server verteilt. Jede Tabelle ist mehrdimensional, wobei jeder Eintrag mithilfe eines Timestamps versioniert wird.

Megastore ist ein auf Bigtable aufbauendes Speichersystem, das die Vorteile von NoSQL-Datenbanken (in erster Linie die Skalierbarkeit) mit den Vorzügen traditioneller RDBMS verbindet. Megastore bietet u.a. konsistente Sekundärindizes, mehrzeilige Transaktionen und konsistente Replikation.



### **2.2.2. Zeilenspeicher**

Die Zeilen aller Datentabellen der Anwender werden gemeinsam in einer einzigen Bigtable Rows gespeichert. Dabei repräsentiert jede Zeile genau eine Zeile in einer Nutzertabelle. Der Primärschlüssel einer Zeile wird intern generiert und besteht aus der Konkatenation der individuellen Tabellen-ID und der jeweiligen Zeile. Dank der guten Skalierbarkeit der Bigtable-Architektur bleibt der Zugriff auf die Rows-Tabelle auch bei Millionen von Anwendertabellen performant, da die Tabelle in einzelne Untertabellen aufgeteilt wird, die wiederum auf verschiedenen Servern abgelegt werden.

Eine Tabellenzeile besteht aus dem Zeilen-Schlüssel sowie je einem Satz indizierter und nicht-indizierter Werte (Properties). Ein Property setzt sich aus einem Tupel aus Property-Name und Property-Wert zusammen. Da folglich jeder Property-Name redundant in der Rows-Tabelle hinterlegt wird, können in den einzelnen Zeilen unterschiedliche Property-Sets (d.h. unterschiedliche Tabellenschemata) gespeichert werden. Für Indizierte Properties wird zum effizienteren Zugriff ein Index aufgebaut.

Standardmäßig werden Property-Werte als Strings gespeichert. Ein Annotationsservice identifiziert automatisch Datentypen wie Zahlen, Datum/Zeit oder Geodaten. Je nach Datentyp schlägt das Google Fusion Tables-Framework passende Visualisierungsformen vor.

### **2.2.3. Schema-Speicher**

Lorem ipsum...

### **2.2.4. View-Speicher**

Lorem ipsum...

### **2.2.5. Kommentar-Speicher**

Lorem ipsum...

## **2.3. Datenabfrage**

Lorem ipsum...

## **2.4. Transaktionen**

Lorem ipsum...

## **2.5. Datenvisualisierung**

Lorem ipsum...

### **2.5.1. Visualisierungsinfrastruktur**

Lorem ipsum...

### **2.5.2. Einbettung**

Lorem ipsum...

## **2.6. Geografische Features**

Lorem ipsum...

### **2.6.1. Google Maps-Infrastruktur**

Lorem ipsum...

### **2.6.2. Einbettung**

Lorem ipsum...

## **2.7. Google Fusion Tables API**

Lorem ipsum...

### **3. Projektvorstellung**

---

Ziel der Spezifikation beinhaltet die Definition der Anforderungen an das zu entwickelnde Datencockpit für die Berliner Weihnachtsmärkte auf Basis von Google Fusion Tables. Neben den Anforderungen an die Applikation sind die Meilensteine, Ziel sowie auch ein einfaches Design / Wireframing enthalten. Die Umsetzung der Applikation "Datencockpit" erfolgt auf Basis dieser Spezifikation.

#### **3.1. Anforderungen**

Sobald die Weihnachtszeit vor der Tür beginnt für viele eine ganz besondere Zeit des Jahres - Weihnachtsmarktzeit. Ende November, mit dem offiziellen Start der Saison schießen sie dann wie Plize aus dem Boden. Es gibt Weihnachtsmärkte in verschiedensten Kategorien, Größen und Ausprägungen, für Kinder, für Jugendliche oder Erwachsene die nur etwas schlendern wollen. Das Überangebot ist dabei sehr unübersichtlich und intransparent. Die Informationen müssen über einzelne Weihnachtsmärkte mühselig zusammen gesucht werden. Interessant sind hierbei vor allem Informationen wie Öffnungszeiten, Anfahrt und ggf. Eintrittspreise. Mehr als diese Informationen sind darüber hinaus meist nicht bekannt.

Das zu entwickelnde Datencockpit soll als zentrale Anlaufstelle für die Suche nach Weihnachtsmärkten dienen. Um die volle Funktionsweise innerhalb des zur Verfügunggestellten Zeitrahmens im Wissenschaftlichen Projekt darstellen zu können, werden nur Berliner Weihnachtsmärkte betrachtet. Allein in Berlin gibt ca. 100 Weihnachtsmärkte. Neben der Bezeichnung "Weihnachtsmärkte" sind Advents-, Winter-, Kiez- aber auch Weihnachtsflohmärkte teil der Analyse und Auswertung im Datencockpit.

Die Funktionalität des Datencockpits unterteilt sich in zwei Hauptbereiche, die Suche und das Einreichen eines neuen Weihnachtsmarktes.

##### **3.1.1. Suche nach Weihnachtsmärkten**

Die Suche ist keine gewöhnliche Volltextsuche sondern letztlich eine Filter, welcher die Ergebnisse auf Basis einer Datenquelle filtert. Die Datenbasis ist

hierbei nicht in einer Datenbank abgelegt sonder in Form einer Google Fusion Table.

Das Datencockpit, was letztlich eine Micro-Website ist, stellt die notwendigen Filtermöglichkeiten zur Verfügung. Nach der Auswahl eines oder mehrerer Kriterien, erfolgt die Ausgabe der gefilterten Ergebnisse. Die einzelnen Kriterien werden dabei mit einem logischen UND verknüpft. Umso mehr Kriterien gleichzeitig ausgewählt werden, desto weniger Ergebnisse werden letztlich zur Verfügung stehen. Die zur Verfügung stehenden Kriterien sollen hierbei ebenfalls in Abhängigkeit angepasst werden, so dass nur mögliche Kriterienkombinationen für die Filterung verwendet werden können.

Aktuelle Filtermöglichkeiten (abhängig von den zu ermittelnden Daten)

- Bezirk
- Öffnungszeiten
- Eintrittskosten
- Sehenswürdigkeiten in der Nähe
- Verfügbare Weihnachtsmärkte an einem bestimmten Tag
- Fahrgeschäfte
- Programme (Kinderprogramm, Live-Musik)
- Sortiment / Kunst

Eine der Stärken von Google Fusion Tables ist unter anderem die Verknüpfung verschiedener Google Fusion Tables. Das heisst, das die eigentlich Datenbasis aus verschiedenen Quellen bestehen kann. Auf diese Weise können vorhandene Informationen verwendet werden und müssen nicht noch einmal in eine eigene Tabelle überführt werden. Für die sinnvolle Verknüpfung benötigt es jedoch ein gemeinsamen Schlüssel. Je nach Kriterium wäre eine Google Fusion Table mit Berliner Sehenswürdigkeiten denkbar. Sofern die PLZ eines Weihnachtsmarkt und einer Sehenswürdigkeit identisch ist, können diese Daten verknüpft werden und dem Benutzer letztlich hilfreiche Informationen zur Sehenswürdigkeiten in der Nähe zur Verfügung gestellt werden.

Die Ausgabe der gefilterten Ergebnisse erfolgt neben der klassischen Listendarstellung auch auf Basis einer interaktiven Karte mit Hilfe von Google Maps.

Eine der Stärken von Google Fusion Tables ist generieren eines eigenen Overlays über einer Google Map. Durch die Angabe von einzelnen Geokoordination können gar ganze Bezirksgrenzen bzw. Weihnachtsmärkte nachgezogen und als eine separate Schicht über der Google Map angezeigt werden. Die Google Map aktualisiert sich dabei nach jeder weiteren Kriterienauswahl. Gewöhnlicher Weise können auf der Google Map einzelne Marker positioniert werden, beispielsweise der Mittelpunkt eines Weihnachtsmarktes. Mit Hilfe von Google Fusion Tables ist jedoch möglich, den genauen Rahmen des jeweiligen Weihnachtsmarktes zu visualisieren. Auf diese Weise erschließt gleich der gesamte Weihnachtsmarkt und auch mögliche Zugänge über benachbarte Straßen.

Um die Weihnachtsmärkte auf einer Karte zu platzieren, müssen die entsprechenden Daten für die Geokoordination ebenfalls vorliegen. Google Maps bietet hierzu verschiedene Möglichkeiten. Das Ziel sollte hierbei sein Weihnachtsmärkte anhand ihrer öffentlichen Adressen direkt auf der Karte platzieren, ohne den Umweg der gewohnten Geokoordination Longitude und Latitude.

Nach dem sich der Benutzer für einen Weihnachtsmarkt entschieden hat ist Verknüpfung an diverse Anfahrtsberechnungssysteme wie Google Maps (Route) oder die Berliner Verkehrsbetriebe (BVG) denkbar.

Neben der Funktion, Benutzer bei der Suche nach bestimmten Weihnachtsmärkten behilflich zu sein, können anhand der Daten anschließend auch verschiedene Diagramme generiert werden. Beispielsweise die Anzahl der Weihnachtsmärkte je Bezirk oder eine Gegenüberstellung der Weihnachtsmärkte mit Fahrgeschäften oder ohne.

### **3.1.2. Einreichung weitere Weihnachtsmärkte**

Neben den oben genannten Möglichkeiten verschiedene Weihnachtsmärkte anhand seiner eigenen Präventionen und Interessen zu filtern gibt es darüber hinaus die Möglichkeit Vorschläge für neue, noch nicht gelistete

Weihnachtsmärkte einzureichen. Gerade in der heutigen Zeit und dem sogenannten Web 2.0 ist die direkte Interaktion mit der Applikation innerhalb bestimmter Benutzergruppen ein wesentlicher Punkt für den Erfolg einer Applikation.

Über eine spezielle Unterseite wird den Benutzer ein spezielles Formular zur Einreichung eines neuen Weihnachtsmarktes zur Verfügung gestellt. Das Formular beinhaltet dabei voraussichtlich folgende Felder:

- Name
- Straße und Hausnummer
- PLZ und Ort
- Bezirk
- Öffnungszeiten
- Eintrittskosten
- Verfügbare Weihnachtsmärkte an einem bestimmten Tag
- Programme (Kinderprogramm, Live-Musik)
- Sortiment / Kunst
- Beschreibung
- Quellen

Nach dem ein Vorschlag für einen neuen Weihnachtsmarkt abgesendet wurde, werden diese Daten mit Hilfe der Google Fusion API direkt in die Google Fusion Table eingefügt. Um ungewünschte Beiträge zu vermeiden, wird es zusätzliche eine Spalte "Status" geben. Die Spalte "Status" besitzt zwei Zustände - show / hide. Ein neuer Datensatz, welcher eingereicht und noch nicht geprüft wurde bekommt automatisch den Status "hide". In der bereits erläuterten Filterung der Daten stehen demzufolge nur Daten zur Verfügung die den Status "show" besitzen und somit einer Prüfung unterzogen wurden.

### **3.1.3. Infrastruktur**

Neben den Anforderungen an die finale Anwendung müssen auch verschiedene technische Anforderungen erfüllt werden:

- Google Drive Benutzer-Accounts für die Google Fusion Tables
- gewöhnlicher Apache Webserver mit PHP
- Integration der Google Fusion API
- GIT-Repository für die gleichzeitige Entwicklung der Anwendung

### 3.2. Ziele

- Ziele (Patrick):  
Wireframing / Design für das Datencockpit (Patrick)
- Berliner Sehenswürdigkeiten:  
<https://www.google.com/fusiontables/DataSource?docid=1HDqaECa6vd-WYiqsXNxpjdgz7A1XLiDyq0MbxMRo>
- Antiquitätenhändler:  
<https://www.google.com/fusiontables/DataSource?docid=1W9Euxyje8G-cwT1FTwfFCNhzbA1bqZfPGAzFIEz4>

## 4. Literatur/Quellen

---

- <http://www.berlin.de/orte/weihnachtsmaerkte/>
- <http://www.mediummagazin.de/magazin-plus/technikipp-so-funktioniert-google-fusion/>
- Berliner Geo-Tags (Bezirksmitten):  
[https://maps.google.com/maps?q=http://toolserver.org/~para/cgi-bin/kmlexport%3Fproject%3Dde%26article%3DListe\\_der\\_Bezirke\\_und\\_Ortsteile\\_Berlins](https://maps.google.com/maps?q=http://toolserver.org/~para/cgi-bin/kmlexport%3Fproject%3Dde%26article%3DListe_der_Bezirke_und_Ortsteile_Berlins)
- Aktuelle Fusion Tables API:  
[https://developers.google.com/fusiontables/docs/v1/getting\\_started?hl=de](https://developers.google.com/fusiontables/docs/v1/getting_started?hl=de)
- deprecated Fusion Tables API:  
[https://developers.google.com/fusiontables/docs/developers\\_guide?hl=de-DE](https://developers.google.com/fusiontables/docs/developers_guide?hl=de-DE)
- Berliner Geo-Tags:  
<http://ags.misterboo.de/#x=13.4246&y=52.5071&z=11>
- Berlin Pankow-Versuch:  
<https://maps.google.de/maps/ms?authuser=0&ie=UTF8&hl=de&oe=UTF8&msa=0&msid=209178059779463884773.0004ce8f590a5735236ce&output=kml>

### 4.1. Conference Papers & Publications<sup>1</sup>

- [Big Data Storytelling through Interactive Maps](#). Jayant Madhavan, Sreeram Balakrishnan, Kathryn Brisbin, Hector Gonzalez, Nitin Gupta, Alon Halevy, Karen Jacqmin-Adams, Heidi Lam, Anno Langen, Hongrae Lee, Rod McChesney, Rebecca Shapley, Warren Shen. Google Inc. [Bulletin of the Technical Committee on Data Engineering, IEEE, June 2012, Vol. 53 No. 2.](#)
- Field Information Management Systems for DNA Barcoding. Deck John, Gross Joyce, Stones-Havas Steven, Davies Neil, Shapley Rebecca, and Meyer Christopher. Chapter 12 in: W. John Kress and David L. Erickson

---

<sup>1</sup> URL: <https://sites.google.com/site/fusiontablestalks/talks>



(eds.), DNA Barcodes: Methods and Protocols, Methods in Molecular Biology, vol. 858

- [Google Earth and Google Fusion Tables in support of time-critical collaboration: Mapping the deepwater horizon oil spill with the AVIRIS airborne spectrometer.](#) Eliza S. Bradley, Dar A. Roberts, Philip E. Dennison, Robert O. Green, Michael Eastwood, Sarah R. Lundeen, Ian B. McCubbin and Ira Leifer. Earth Science Informatics, 17 September 2011.
- [Fusion Tables: new ways to collaborate on structured data.](#) Jonathan Goldberg Kidon, Massachusetts Institute of Technology. Dept. of Electrical Engineering and Computer Science, 2010. 61 pages.
- [Google Fusion Tables: Web-Centered Data Management and Collaboration.](#) Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen, Jonathan Goldberg-Kidon. Proceedings of SIGMOD 2010 (Industrial Track), Indianapolis, Indiana.
- [Google Fusion Tables: Data Management, Integration and Collaboration in the Cloud.](#) Hector Gonzalez, Alon Y. Halevy, Christian S. Jensen, Anno Langen, Jayant Madhavan, Rebecca Shapley, Warren Shen. Proceedings of the Symposium on Cloud Computing, 2010 (Industrial Track), Indianapolis, Indiana.

#### **4.2. Presentations & workshops by the Fusion Tables team<sup>2</sup>**

- Data Boot Camp using Google tools, LA Hackathon on Immigration, 8-9 Dec 2012
- Data sharing, visualization and app building with Fusion Tables, Open Developer Week Dec 2012
- Find data and Merge faster, a free training over Google Hangouts, October 2012.
- TechRaking: CIR and Google, April 12, 2012, "Tips and Tricks: Fusion Tables and more"
- Investigative Reporters & Editors CAR conference, "Getting the most out of Fusion Tables" demo and "Advanced Fusion Tables" hands-on workshop, Feb 2012

---

<sup>2</sup> URL: <https://sites.google.com/site/fusiontablestalks/talks>

- A data tools and visualization workshop in January 2012 in Nairobi Kenya
- Silicon Valley Code Camp, October 2011
- Online News Association, September 2011
- Managing and Visualizing your geospatial data with Fusion Tables, Google I/O video, May 2011
- Data 2.0 conference workshop April 2011
- Silicon Valley GTUG, April 2011
- Where 2.0 workshop, April 2011
- Investigative Reporters & Editors CAR conference, NICAR demo & workshops February 2011
- Association of American Geographers, "Visualize gobs of data with Fusion Tables" April 2010
- Google Earth Outreach, Train the Trainers event 2.5 hour training and 1 hour training, The International Livestock Research Institute , Nairobi, Kenya Nov 2009
- Google Geo-Public sector partnership event, UN, New York, Nov, 2009
- Google Geo-Public sector partnership event, Washington DC, Oct. 2009

## **ToDoS / Meilensteine / WIP**

1. Einarbeitung
  - (a) Google Fusion Tables (Manuel)
  - (b) Google Maps (Manuel)
  - (c) Open Data Kit (Manuel)
2. Spezifikation / Lastenheft
  - (a) Anforderungen (Patrick)
  - (b) Meilensteine (Patrick)
  - (c) Ziele (Patrick)
  - (d) Wireframing / Design für das Datencockpit (Patrick)
3. Vorbereitung
  - (a) Datenbeschaffung
    - i. Geo-Koordinaten für Berliner Bezirksgrenzen
    - ii. Weihnachtsmärkte in Berlin
      - A. Ort / Geodaten
      - B. Öffnungszeiten
      - C. Evtl. Detailinformationen (Glühweinpreise, Fahrgeschäfte, etc.)
  - (b) Infrastruktur einrichten
    - i. VCS / GitHub
    - ii. Entwicklungsumgebung / HTML / etc.
    - iii. Website / Webspace einrichten
4. Umsetzung
  - (a) GFT -Struktur anlegen und mit Daten befüllen
  - (b) Kombination GFT mit Google Maps
  - (c) HTML-Prototyp für Datencockpit
  - (d) SQL, neue Einträge in GFT [Open Data Kits]
5. Dokumentation der verwendeten Technologien
  - (a) GFT
  - (b) Verbindung mit Maps
  - (c) Open Data Kit
  - (d) Schwierigkeiten, Herausforderungen, etc.
6. Abschlusspräsentation (Peka Kucha-Stil)