



**memonic** 20091201

## Neue Technologien für bessere Websites



Internet Briefing

1. Dezember 2009

Dr. Dorian Selz  
Co-Founder, CEO

# Remix the Web



## Einfach zugängliche Daten:

- Karten
- Fotos
- Events
- News
- Produkte
- Suche
- ... jedwelche Daten

memonic

3

memonic is a trademark of Nektoon Inc. – confidential & proprietary – © Nektoon Inc. 2009



## Zum Beispiel: Schweinegrippe-Tracker

- Informationen von der WHO
- Kartenmaterial und Infrastruktur von Google
- Motivation und Zeit von Anwendern
- Resultat: Brandaktuelle Informationen

memonic

4

memonic is a trademark of Nektoon Inc. – confidential & proprietary – © Nektoon Inc. 2009





## Zu einem erfolgreichen Remix gehören:

- Zugang zu Daten
- SOA (Service-oriented Architecture) hilft
- Offene und dokumentierte API
- Gescheites Modell für den Gebrauch der API
  - Was darf die Drittpartei mit den Daten machen
  - Beispiel SBB: Freigeben macht Sinn; das eigentliche Geschäft ist der Ticketverkauf
  - Beispiel Google: Offene API; aber ab bestimmtem Erfolg kann Werbung eingeblendet werden

memonic

5

memonic is a trademark of Nektoon Inc. – confidential & proprietary – © Nektoon Inc. 2009



## Wie sind eigene Applikationen auszurichten?

- Service encapsulation: Die Services müssen für sich stehen können.
- Service loose coupling – Kaum (besser: keine) gegenseitigen Abhängigkeiten
- Service contract – Gemeinsamer Kommunikationsstandard
- Service abstraction – Logik wird gegenüber "ausen" versteckt
- Service reusability – Einzelne kleine Dienste die wiederverwendet werden können
- Service composability – Dienste können automatisch gebaut werden
- Service autonomy – Jeder Dienst tut genau Etwas gut
- Service discoverability – Die Dienste werden zugänglich gemacht

memonic 20091201

6

memonic is a trademark of Nektoon Inc. – confidential & proprietary – © Nektoon Inc. 2009





## Links

---

- [http://en.wikipedia.org/wiki/Mashup\\_\(web\\_application\\_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid))
- <http://www.programmableweb.com/mashups>
- <http://www.webmashup.com/>
- <http://mashupawards.com/>
- [http://en.wikipedia.org/wiki/Service-oriented\\_architecture](http://en.wikipedia.org/wiki/Service-oriented_architecture)



## Social Graph



## Soziale Netzwerke: Definition

- Erstellen eines öffentlichen oder semi-öffentlichen Profils
- Benennen von und verbinden mit andern Nutzern mit denen mich etwas verbindet
- Einsehen können der Verbindungen von andern Nutzern



## Fazit Social Graph & Friend Feeding

- Das soziale Web ist auch für Firmen relevant
- Es braucht ein Umdenken: Zentrale PR Abteilungen werden dezentralisiert
- Unbedingt Beiträge über die eigene Firma begleiten
- Die Vorteile überwiegen die Nachteile





## Links

---

- Realtime Web
  - <http://www.twitter.com>
  - <http://www.yammer.com>
  - <http://beta.twittervision.com>
  - [http://de.wikipedia.org/wiki/Extensible\\_Messaging\\_and\\_Presence\\_Protocol](http://de.wikipedia.org/wiki/Extensible_Messaging_and_Presence_Protocol)
  - [wave.google.com/](http://wave.google.com/)
- Observation
  - <http://www.buzzient.com>
  - <http://www.socialmention.com>



## Rich Browsers



## Definition eines modernen Browsers

- Consistent and transparent W3C Web Page rendering
- Intuitive and responsive user Interface
- Beginning support for advanced HTML 5 and CSS3
- Fast Javascript Performance

memonic

13

memonic is a trademark of Nektoon Inc. – confidential & proprietary – © Nektoon Inc. 2009



## Jedem Browser seine eigene Layout-Engine

- Trident: Internet Explorer (Microsoft)
- Gecko: Firefox, et al. (Mozilla)
- Presto: Opera Browser (Opera)
- KHTML: Konqueror
- WebKit: Safari (Apple), Nokia & Google Chrome

memonic

14

memonic is a trademark of Nektoon Inc. – confidential & proprietary – © Nektoon Inc. 2009





## Fazit: Rich Browsers

---

- Hin und Her zwischen Client und Server bei der Aufteilung der Logik
  - Damals: VT400 (Logik auf Serverseite)
  - Später: PC (Logik auf Clientseite)
  - Web 1.0: Logik auf Serverseite
  - Web 2.0: Logik geteilt zwischen Server und Client
  - Web 3.0: ?
- Wie weiter?
  - Browser(IN-)kompatibilitäten verursachen Schwierigkeiten
  - Ausrichtung der eigenen Applikationen auf Rich Browsers



## Links

---

- [http://en.wikipedia.org/wiki/Comparison\\_of\\_layout\\_engines\\_\(HTML\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(HTML))
- [http://en.wikipedia.org/wiki/Rich\\_Internet\\_application](http://en.wikipedia.org/wiki/Rich_Internet_application)
- <http://code.google.com/p/richbrowser/>
- <http://code.google.com/p/o3d/>
- <http://gears.google.com>
- <http://code.google.com/intl/en/apis/o3d/>
- <http://www.webkit.org>





## Skalierbare Infrastruktur

### Shared Nothing Architecture

- Mit der Hardware skalieren
- RESTful
- Vorteile von HTTP ausnützen





## Automatisierung der Abläufe

- Server Setup & Deployment
- Continuous Integration
  - Kontinuierliches Building/testing von Software Projects
    - Hudson provides an easy-to-use so-called continuous integration system, making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build.
  - Monitoring von Drittjobs (z.B. Cron Jobs)
    - Hudson keeps those outputs and makes it easy for you to notice when something is wrong.
- Testing



Cloud Computing ist ein  
Missverständnis



## Cloud Compting

---

- Bezug von Storage, Database, Messaging, Processing aus der „Cloud“
- Amazon als Vorreiter
- Heute eine Vielzahl von Anbietern
- De-facto Standardisierung je rund um Amazon (Sun, Rackspace, etc.) und Google



## Virtualisierung von fast allem

---

- Kaum eigene Infrastruktur, „Pay-As-You-Go“
- Content-Delivery-Networks (CDN) und Dynamic Loadbalancing ermöglichen genaue Steuerung auch von Nachfragespitzen
- Gebrauch von „Fertiglösungen“; Andere lösen unsere Probleme (Server Setup, Maintenance, etc.)
- Die eigenen Dienste nahe beim Kunden anbeiten





## Fazit skalierbare Infrastruktur

---

- Nächste Etappe im Bau von Web-Applikationen
- Klare (massive) Kostenvorteile
- Noch in einer frühen Phase: In Teilen noch nicht „industriereif“ aber jetzt ist der Einstiegszeitpunkt
- Wiederholung der Motorisierung der Wirtschaft:
  - Damals: Jede Fabrik ihre eigene zentrale Dampfturbine
  - Heute: Motoren direkt dort wo kinetische Energie gebraucht wird, gemeinsame Elektrizitätsversorgung



## Links

---

- Shared Nothing Architecture
  - [http://en.wikipedia.org/wiki/Shared\\_nothing\\_architecture](http://en.wikipedia.org/wiki/Shared_nothing_architecture)
  - [http://de.wikipedia.org/wiki/Representational\\_State\\_Transfer](http://de.wikipedia.org/wiki/Representational_State_Transfer)
  - <http://startwerk.ch/2009/08/11/bauanleitung-fuer-ein-startup-share-nothing-architecture/>
- Automation
  - <http://reductivelabs.com/products/puppet>
  - <http://hudson-ci.org>
  - <http://www.atlassian.com/software/bamboo>
- Virtualization / Cloud Computing
  - [http://de.wikipedia.org/wiki/Cloud\\_Computing](http://de.wikipedia.org/wiki/Cloud_Computing)
  - [http://de.wikipedia.org/wiki/Content\\_Distribution\\_Network](http://de.wikipedia.org/wiki/Content_Distribution_Network)
  - <http://aws.amazon.com>
  - <http://www.akamai.com>
  - <http://www.rackspacecloud.com>
  - <http://www.sun.com/solutions/cloudcomputing/index.jsp>
  - <http://www.cloudsuisse.org/>





**memonic**

Danke für Eure Zeit!

memonic by Nektoon Inc.

Nektoon Inc.  
Feldstrasse 133  
CH-8004 Zürich

+41 79 419 67 05  
dorian@nektoon.com  
www.nektoon.com

