Pushing Enterprise Software to the Next Level
Self-contained Web Applications on In-Memory Platforms

Michał Nosek
Starcounter AB
Who am I?

- Michał Nosek
  Software Engineer, Technical Sales Engineer – Starcounter
  http://starcounter.com

- Github: mmnosek
  LinkedIn: https://www.linkedin.com/in/mmnosek
  E-mail: michal@starcounter.com
  Twitter: @mmnosek
On Today’s Agenda

01
Setting the Stage
RAM Memory
Modern WEB
SCS Architecture

02
In-Memory Application Platform
Architecture
Single App
Integration
Demo
Future
## Enterprise Software of Today

<table>
<thead>
<tr>
<th><strong>Monolith</strong></th>
<th><strong>Micro-Services</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad maintainability</td>
<td>Orchestration</td>
</tr>
<tr>
<td>Long builds</td>
<td>Eventual consistency</td>
</tr>
<tr>
<td>Technology lock-in</td>
<td>Communication problems</td>
</tr>
<tr>
<td>Long TTM</td>
<td>Complexity</td>
</tr>
<tr>
<td>Poor scalability</td>
<td></td>
</tr>
</tbody>
</table>
Wirth’s law

“What Intel giveth, Microsoft taketh away.”

“What Andy giveth, Bill taketh away”
On Today’s Agenda

01
Setting the Stage
RAM Memory
Modern WEB
SCS Architecture

02
In-Memory Application Platform
Architecture
Single App
Integration
Demo
Future
RAM Prices

Price of 1MB in USD over time

USD

1000,000000
100,000000
10,000000
1,000000
0,100000
0,010000
0,001000
0,000100
0,000010
0,000001


Year

https://jcmit.net/memoryprice.htm
Conventional

In-Memory

![Diagram showing the difference between Conventional and In-Memory systems]

- **Conventional**
  - Apps 1 and 2 use Heap memory.
  - Data access goes through DBMS, which interacts with Disk and Log.

- **In-Memory**
  - Apps 1 and 2 use Heap memory.
  - Data access is direct from Storage, bypassing DBMS.
<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Getting faster</td>
<td>▪ Communication isn’t faster</td>
</tr>
<tr>
<td>▪ Better utilised by modern CPUs</td>
<td>▪ It’s not durable</td>
</tr>
<tr>
<td></td>
<td>▪ Not getting cheaper anymore?</td>
</tr>
</tbody>
</table>
On Today’s Agenda

01
Setting the Stage
RAM Memory
Modern WEB
SCS Architecture

02
In-Memory Application Platform
Architecture
Single App
Integration
Demo
Future
## Pros and Cons

### Pros
- Ubiquitous (no native, separate process)
- Semantics (content) vs Presentation
- Modularity as priority (reusability)

### Cons
- Still not implemented everywhere
- Global scope (one app can break something in another)
- Online requirement
On Today’s Agenda

01 Setting the Stage
   RAM Memory
   Modern WEB
   SCS Architecture

02 In-Memory Application Platform
   Architecture
   Single App
   Integration
   Demo
   Future

source: scs-architecture.org
SCS Architecture

User interface
Business logic
Persistence
If you cut and wrap every domain in a separate web application
An SCS contains its own user interface, specific business logic and separate data storage.
SCS Architecture
Pros and Cons

Pros
- Modularisation
- Maintainability
- Loose coupling

Cons
- Integration
- Common look and feel
- Inconsistency
On Today’s Agenda

01
Setting the Stage
RAM Memory
Modern WEB
SCS Architecture

02
In-Memory Application Platform
Architecture
Single App
Integration
Demo
Future
In-Memory Application Platform

For Building Self-Contained Systems
General Platform Architecture

- **Front-end Framework**: React, Polymer
- **Communication**: Palindrom - REST, Web Sockets
- **Application**: View Models, Entities, App Logic
- **In Memory App Platform**: Mapping, Persistence, Queries

Starcounter
Traditional Stack vs Starcounter Stack

**TRADITIONAL**
- front-end framework
- client-side logic
- network
- services + contracts
- application code
- O/R Mapper
- DBMS

**STARCOUNTER**
- front-end framework
- network
- view models
- application
- in-memory app platform
Data Storage

- In-Memory database
- ACID compliant
- Snapshot isolation
- Flexible
VMDBMS

U.S. Patent No. 8,266,125
Business Logic

- Polyglot
- Simplified
- Platform-agnostic
- Real-time
User Interface

- Web native
- Web socket communication
- Design agnostic
- Thin
Demo:
Simple SCS app
Integration: Data Level
UI A  UI B  UI C
App A  App B  App C
Model A  Model B  Model C
Mapper

Starcounter
Integration: UI Level
<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firstname</td>
<td>Stephen</td>
</tr>
<tr>
<td>Lastname</td>
<td>Fry</td>
</tr>
<tr>
<td>Primary Email</td>
<td><a href="mailto:Stephen.Fry@Starcounter.com">Stephen.Fry@Starcounter.com</a></td>
</tr>
<tr>
<td>Primary Phone</td>
<td>+46 71 234 36 78</td>
</tr>
<tr>
<td>Shipping Addr</td>
<td>Dammmsugärvägen 50</td>
</tr>
<tr>
<td></td>
<td>123 45</td>
</tr>
<tr>
<td></td>
<td>Panchrrillistan</td>
</tr>
<tr>
<td></td>
<td>County</td>
</tr>
</tbody>
</table>
### Customer Information

**Firstname:** Stephen  
**Lastname:** Fry  
**Email:** Stephen.Fry@Starcounter.com  
**Phone:** +46 71 234 56 78  
**Address:** Dammssgården 53, 123 45, Panchristian

### Subscriptions

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Quantity</th>
<th>Start Date</th>
<th>Next Delivery</th>
<th>Discount (%)</th>
<th>Frequency (Days)</th>
<th>Price (SEK)</th>
<th>Active</th>
<th>Campaign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexovital</td>
<td>1</td>
<td>2018-02-21</td>
<td>2018-02-21</td>
<td>0</td>
<td>1</td>
<td>500.00</td>
<td>✔</td>
<td>None</td>
</tr>
<tr>
<td>Endurance</td>
<td>10</td>
<td>2018-02-21</td>
<td>2018-02-21</td>
<td>0</td>
<td>1</td>
<td>1500.00</td>
<td>✔</td>
<td>None</td>
</tr>
<tr>
<td>Endurance</td>
<td>5</td>
<td>2018-02-22</td>
<td>2018-02-22</td>
<td>0</td>
<td>1</td>
<td>750.00</td>
<td>✔</td>
<td>None</td>
</tr>
<tr>
<td>Flexovital</td>
<td>1</td>
<td>2018-02-27</td>
<td>2018-02-27</td>
<td>0</td>
<td>28</td>
<td>198.00</td>
<td>✔</td>
<td>None</td>
</tr>
</tbody>
</table>
Stephen Fry
Customer Id:

First Name: Stephen
Last Name: Fry
Email: Stephen.Fry@Starcounter.com
Phone: +46 71 234 36 78
Shipping Address: Dammsugavgatan 53 123 45, Puchallistan

Subscriptions:

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Quantity</th>
<th>Start Date</th>
<th>Next Delivery</th>
<th>Discount (%)</th>
<th>Frequency (Days)</th>
<th>Price (SEK)</th>
<th>Active</th>
<th>Campaign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexovital</td>
<td>1</td>
<td>2018-02-21</td>
<td>2018-02-21</td>
<td>0</td>
<td>1</td>
<td>500.00</td>
<td></td>
<td>None</td>
</tr>
<tr>
<td>Endurance</td>
<td>10</td>
<td>2018-02-21</td>
<td>2018-02-21</td>
<td>0</td>
<td>1</td>
<td>1500.00</td>
<td></td>
<td>None</td>
</tr>
<tr>
<td>Endurance</td>
<td>5</td>
<td>2018-02-22</td>
<td>2018-02-22</td>
<td>0</td>
<td>1</td>
<td>750.00</td>
<td></td>
<td>None</td>
</tr>
<tr>
<td>Flexovital</td>
<td>1</td>
<td>2018-02-27</td>
<td>2018-02-27</td>
<td>0</td>
<td>28</td>
<td>198.00</td>
<td></td>
<td>None</td>
</tr>
</tbody>
</table>
Stephen Fry

Customer Id.

Firstname: Stephen
Lastname: Fry

Primary Email: Stephen.Fry@Starcounter.com

Primary Phone: +46 71 234 06 78

Address:
Dammisugarvägen 53, 123 45, Puchrullistan

Subscriptions:

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Quantity</th>
<th>Start Date</th>
<th>Next Delivery</th>
<th>Discount (%)</th>
<th>Frequency (Days)</th>
<th>Price (SEK)</th>
<th>Active</th>
<th>Campaign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexovital</td>
<td>1</td>
<td>2018-02-21</td>
<td>2018-02-21</td>
<td>0</td>
<td>1</td>
<td>500.00</td>
<td>✔</td>
<td>None</td>
</tr>
<tr>
<td>Endurance</td>
<td>10</td>
<td>2018-02-21</td>
<td>2018-02-21</td>
<td>0</td>
<td>1</td>
<td>1500.00</td>
<td>✔</td>
<td>None</td>
</tr>
<tr>
<td>Endurance</td>
<td>5</td>
<td>2018-02-22</td>
<td>2018-02-22</td>
<td>0</td>
<td>1</td>
<td>750.00</td>
<td>✔</td>
<td>None</td>
</tr>
<tr>
<td>Flexovital</td>
<td>1</td>
<td>2018-02-27</td>
<td>2018-02-27</td>
<td>0</td>
<td>28</td>
<td>198.00</td>
<td>✔</td>
<td>None</td>
</tr>
</tbody>
</table>
# Outcomes

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modularisation</td>
<td>Integration</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Different look and feel</td>
</tr>
<tr>
<td>Loose coupling</td>
<td>Inconsistency</td>
</tr>
<tr>
<td>Full and easy integration</td>
<td>Platform lock-in?</td>
</tr>
<tr>
<td>Common look and feel</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td></td>
</tr>
<tr>
<td>Modules / Variants</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Shell module</td>
<td>Shell and container Styling</td>
</tr>
<tr>
<td>Shell, Extended</td>
<td>Extended, w/ &quot;dust wheel&quot;</td>
</tr>
<tr>
<td>Shell, Short</td>
<td>Short, Compact</td>
</tr>
<tr>
<td>Shell, Long</td>
<td>Long, Standard</td>
</tr>
<tr>
<td>Dust flap module</td>
<td>Dust flap One-way-valve</td>
</tr>
<tr>
<td>Dust flap, long life</td>
<td>Long Life, Silicone</td>
</tr>
<tr>
<td>Dust flap, std</td>
<td>STD, Black rubber</td>
</tr>
<tr>
<td>Charger connect module</td>
<td>Charge plug and Charge so...</td>
</tr>
<tr>
<td>Dock module</td>
<td>Wall mounting, Docking Uni...</td>
</tr>
<tr>
<td>PCB module</td>
<td>Printed circuit board, electro...</td>
</tr>
<tr>
<td>Nozzle module</td>
<td>Nozzle (fixed)</td>
</tr>
<tr>
<td>Control module</td>
<td>On/Off switch, Container rel...</td>
</tr>
<tr>
<td>Container module</td>
<td>Container for dust collection...</td>
</tr>
<tr>
<td>Handle module</td>
<td>Handle, used for Styling</td>
</tr>
<tr>
<td>Charger module</td>
<td>Power Adopter (transformer)</td>
</tr>
<tr>
<td>Base module</td>
<td>Shell base</td>
</tr>
<tr>
<td>Filter module</td>
<td>Dust filter (Detachable)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>USD</td>
<td>-</td>
</tr>
<tr>
<td>Brand</td>
<td>-</td>
</tr>
<tr>
<td>Region</td>
<td>-</td>
</tr>
<tr>
<td>VDI</td>
<td>S</td>
</tr>
<tr>
<td>Surface texture</td>
<td>V</td>
</tr>
<tr>
<td>Size of inset</td>
<td>B</td>
</tr>
<tr>
<td>Styling shape</td>
<td>2</td>
</tr>
<tr>
<td>Air Flow</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Module Variant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCP</td>
<td>6</td>
</tr>
</tbody>
</table>
**Full-Stack Benchmark**

- 1.5 mln. accounts, 500 K remote clients transfer.
- Money between accounts (5%) and read totals (95%).
- Transfer and read operations are mixed randomly.
- Starcounter on .NET (1 x EC2 c3.8xlarge): 1 M OPS.
- MariaDB Galera Cluster 5 nodes with Node.js app server (5 x EC2 c3.2xlarge, EBS root volume and high network throughput, stored procedures): 55 K OPS.
- Ratio suffers for MariaDB doing more writes.

**Storage Engine Benchmark**

- YCSB load 5% writes, 95% reads.
- 1 x E5-2680v2, 1 machine (10/20 cores/threads).
- 8 threads: 3.5 mln. Ops/sec.
- 16 threads: 5.4 mln. Ops/sec.
- c3.8xlarge – 60 GiB RAM, 32 vCPUs
- c3.2xlarge – 15 GiB RAM, 8 vCPUs
- https://www.ec2instances.info/
On Today’s Agenda

01
Setting the Stage
RAM Memory
Modern WEB
SCS Architecture

02
In-Memory Application Platform
Architecture
Single App
Integration
Demo
Future
Currently vs Future

Current
- Registers
- Cache
- Main Memory
- Solid State Disk
- Magnetic Disk

Future
- Registers
- Cache
- Main Memory
- Non Volatile Memory
- Solid State Disk
- Magnetic Disk
Starcounter in the Future

![Diagram showing Starcounter's future architecture with a CPU, Volatile DRAM, Non-Volatile DRAM, Write/Read processes, Delta Store (incremental updates), Merge, and Main Store (batch update).]
Enterprise Software of Tomorrow

- Simplified
- Near real-time
- Easy to maintain
- Reusable/modularised
- Fully web-based
- Fast data
- HTAP or HOAP
THANK YOU!
Questions?