Marc Weber - IT Spezialist

Tel.: +49 176/60032282 · E-Mail: marco-oweber@gmx.de, Im Tannhörnle 4/1 78052 Villingen-Schwenningen

Lebenslauf

Angaben zur Person

geboren:	20.1.1984, Schwenningen, Deutschland
Familienstand	ledig
Staatsangehörigkeit	Deutsch

Ausbildung

1990-1994	Grundschule Pfaffenweiler (bei Villingen-Schwenningen)
Ende 4. Klasse	musikalische Laufbahn: Tenorhorn, Trompete im Musikverein, Leistungsabzeichen Bronze, Silber, Gold des BDB, habe damals in 4 verschiedenen Musikgruppen gespielt
1994-2002	Gymnasium am Romäusring (naturwissenschaftlicher Zug) mit Abschluss Abitur
1998-1999	Erfolgreiche Teilnahme an der ersten Runde des 17. Bundeswettbewerb Informatik
2003-2006	Physik + Mathe Studium ohne Abschluss an der Universität Konstanz Erworbene Scheine: Analysis I+II, lineare Algebra I, Chemie I, Integrierter Kurs theoretische und praktische Physik I + II Studien1. Dann Fokussierung aufs Programmieren.

Marc Weber - IT Spezialist

Tel.: +49 176/60032282 · E-Mail: marco-oweber@gmx.de, Im Tannhörnle 4/1 78052 Villingen-Schwenningen

Technologien, mit denen ich bereits am Computer gearbeitet habe

kompilierte Sprachen:
Java,Scala, C/ C++, JavaScript, VBA, VB, .net, Pascal, Lazarus, Delphi (1-7),
Ocamml, Haxe, Haskell, impredicative.com/ur, ML

Scriptsprachen:
VimL, Nix, Bash, ZSH, Python, Ruby, PHP, TCL

Tools:
Shell Scripting etc, Vim-Experte (modaler Editor)

Datebanken:
SQL (MySQL, sqlite, Postgresql, ..)
NoSQL (etwas Rethinkdb, Arangodb Erfahrung)

Technologien / Stichwörter
Haxe, BDD, TDD, Sockets, Threads, Locking,
SQLServer, Postgis, Qgis, Mootools, Jquery, CSS, HTML, gcc, ghc, OpenGL,
Microcontroller, wmii, Selenium, phantomjs, PayPal, RubyForge, PyPi, Etherpad,
Hackage, RayTracers, Blender, Gimp-Scripting, Office, Linux, Windows,
Autohotkey, Browser-Plugins für Opera, Chrome, Firefox, u.v.m

Marc Weber - IT Spezialist

Tel.: +49 176/60032282 · E-Mail: marco-oweber@gmx.de, Im Tannhörnle 4/1 78052 Villingen-Schwenningen

Beruflicher Werdegang

während Schulzeit	5 Wochen Datenbank-Softwareentwicklung mit VB, ADODB und Access in den Sommerferien beim Ingenieur Büro Günther Fluck www.ibgf.de.
nach Studium	
01.2006 - 02/2006	Praktikum als Java und .net Entwickler bei http://www.skillworks.de
09.2006 - 12/2006	Zeitarbeit ber Personalfirma Soppa.
04.2007 - Heute	Selbständigkeit als EDV-Diensleister bzw Programmierer.

Mitarbeit an Projekten (2007-heute)

04.2007 - Heute	Aus Überzeugung bin Ich aktiver Comitter beim Projekt NixOS: NixOS ist ein Linux-System, da s auf dem Paket-Manager Nix basiert. Er wird „pure“ oder rein genannt, weil er auf einer Idee basiert: das ganze System kann durch eine Datei konfiguriert werden. Alte Konfigurationen bleiben verfügbar bis man den Garbage-Collector ausführt. Weil das Projekt noch sehr jung war, musste ich für weit über 100 Pakete Installations-Skripte schreiben Dabei sind auch automatisierte Tools zum erstellen von Paketen für Haskell, Ruby und Python Packages entstanden, die als Quelle Hackage, RubyForge oder PyPi verwenden. NixOS benutze ich als tägliches Desktop System und als Hostingplatform für die Projekte die ich aus meiner Freelancer Tätigkeit betreue. Weitere Informationen: http://nixos.org, meine Commits sind einfach durch Suche nach meiner Emailadresse zu finden, auch http://github.com/MarcWeber/nixpkgs enthält einige weitere Patches, wie php-fpm Integrationen uvm. Auf der gleichen Idee basiert der VIM-Addon-Manager VAM, den ich für die VIM Community (und aus eigenem Interesse) geschrieben habe.
2008 - heute	Neuimplementation eines **Bookingsystems** für befreundete Firma **Webkos Mönchweiler.** Eingesetzte Technologien: MySQL, PHP, Ajax = asynchrone JavaScript Kommunikation mit dem Mootools JavaScript Framework Das System ist bei ca 25 Kunden im Einsatz. Das Projekt entählt ca 30 Untermodule die vom einfachen Buchen, der Gewinnermittlung, Todo Listen, Rechteverwaltung bis hin zur Rechnungserstellung alle Wünsche abdecken.

Marc Weber - IT Spezialist

Tel.: +49 176/60032282 · E-Mail: marco-oweber@gmx.de, Im Tannhörnle 4/1 78052 Villingen-Schwenningen

2011	**haml-to-php.com entwickelt**. **Erkenntnis**: Mit Libraries ist es schwer Geld zu verdienen. Man verdient nur Geld mit fertigen Produkten.
2008 - 2012	**Support** eigene Projekte auf http://github.com/MarcWeber Kennenlernen von neuen Programmiersprachen wie Haxe und urweb Hilfe bei **Notfallrettung Firma Webkos Mönchweiler wegen andauernden DDOS Attacken** (4 Wochen)– Hilfe bei der Strategiefindung zur Lösung (=> temporärer Umzug auf Amazon, Cloud Computing etc, aber ohne Wettrüsten, denn Angreifer bezahlen meist nichts).
2012 bis heute	Verschiedene **Freelance-Arbeiten**. Auftragsarbeiten für verschiedene Firmen wie agree-it.de, Evaluation von Projekten..
2014 bis heute	Auftragsargbeiten für agree-it.de, JobVers GmbH, Erstellung einer Versicherungsmakler Platform inkl. Scrapings vieler Versicherungsmaklerseiten.
2015 / 2016	Auftragsarbeiten für Paranoid Internet GmbH Deutschland, z.B holifestival.de (vorgegebenes Design), A/B Testingläufe, eigene Auswertungen & User Tracking u.v.m. Jobvers.de => Bestehende Programme erweitert und verknüpft.
2016	Für webkos.de: SMS Gateway implementiert – Watrungs und Erweiterungsarbeiten jobvers GmbH / webkos.de
2016	Backend + cordova App Video + Bilder anzeigen mit Deeplinking
2017	Auftragsarbeiten für Webkos und JobVers.de / Auseinandersetzung mit Cryptocoins um die Wichtigeit zu verstehen und weiteres
2017 – heute	Verschiedene Ideen für Investment vorbereitet ,aber keinen Investor gefunden. Für Bestandskunden gearbeitet. Einarbeitung TypeScript. Angefangen neue Packaging Systeme für TypeScript zu schreiben, weil Ich hier die Zukunft sehe, abgesehen vom schreiben neue Programmiersprachen, was aber 2+ Jahre dauern würde ;-(TypeScript ist ein guter Kompromiss fürs Web aus meiner Sicht was aktuelle Technology anbelangt. Wachsende Erkenntnis dass Ich in der Zukunft gerne Legacy & Future verbinden würde was Poly also Integration der neuen und der alten Programmiersprachen bedeutet.

Marc Weber - IT Spezialist

Tel.: +49 176/60032282 · E-Mail: marco-oweber@gmx.de, Im Tannhörnle 4/1 78052 Villingen-Schwenningen

Ich habe vercshiedene Ideen wie folgende nach dem Leanstartup Prinzip venrsucht umzusetzen bin aber im Verkauf gescheitert:

Rotating cutter Knife
https://www.youtube.com/watch?v=cgRNzsfV7Lc&t=237s
Fixing von PythonAutohotkey Library Windows.
FernsteuerunWarum 20 Jahre ? Die Ziele zu erreichen wird 10 Jahre dauern, bis Communities übernehmen wirds nochmal 10 Jahre dauern
Ich bin ein bißchen müde immer neue Tools und Herausforderungen meistern zu müssen, und dennoch gegen Wände zu rennen. Egal wie groß der Hype im Internet dargestellt ist.
Beispiele

Nim: Return Types angeben kostet Zeit. Autor will nichts dran ändern.
Bidirectional Typing (wie z.B. in Haskell seid langem Standard ist wirds nicht geben)
Haxe: map(_ + 1) -> short lambdas -> Patch vom Autor abgelehnt
Rust -> Es gibt einen Grund warum es den Alloy Fork gibt (GC integriert).
Gleiches Problem wie bei Nim. Zusätzlich Shared Memory nutzen mit anderen Prozessen eventuell nicht so einfach. D.h. wenn man 100 GB im RAM und bischen Code schnell ändern will muss man alles neu laden.
TypeScript: Spart Zeit weil return Types oft von alleine erkannt werden.
Jedoch execve geht nicht so einfach oder macros fehlen (Ja bun aber .. IDE?)
Java/Koltlin/Dart/Haxe: OO lässt bestimmte Optimierungen nicht zu. GC ist für einige Anwendungen schlecht. Es gibt einen Grund warumt man bei Java Abhandlungen findet welche Art von GC bei welcher Anwendung mit welcher Konfiguration.
Problem gibts bei Rust nicht.
Zig löst das Shared Memory Problem (Allocator immer übergeben), aber eventuell fehlen noch Libraries und ist noch sehr neu

Deswegen möchte Ich eine Entwicklungsumgebung schaffen, welche verschiedene Execution Contexts
erlaubt, verschiedene Memory-Management Techniken je nachdem was wichtig für eine Aufgabe ist.
Und einen graduellen Übergang von kompiliert, interpretiert, jitted, dynamisch erlaubt und gleichzeitig SubTyping wie TypeScript integriert und bidirectionales Typing wie Haskell.

Execution Contexts Beispiele:
Webassembler > Browser > Server > DB
Verschiedene Schichten in Browser Extensions

Tel.: +49 176/60032282 · E-Mail: marco-oweber@gmx.de, Im Tannhörnle 4/1 78052 Villingen-Schwenningen

Oder verschiedene Programmiersprachen
Transparente API Calls wie bei RakkasJS (runSSQ)

Shell Befehl Integration bzw Auflösung des Unterschieds Library / Exceutable

https://github.com/MarcWeber/gradualc

Während Ich Nixos Linux liebe, fehlen ein paar abstraktere schichten ähnlich requirements.txt (Python) welche es erlauben Abhängikeiten in einer Cross-Platform Cross Language Art und Weise definieren zu können, so dass als Nebenprodukt Entwicklungsumgebungen gebaut werden können und Pakete für Nixpkgs, chocolately, brew, conda, mamba, gentoo abgeleitet.

https://github.com/MarcWeber/it-glue

Die Probleme fallen mir immer wieder auf die Füße, deswegen fällt es mir schwer loszulassen. CNC
Python Programm um CNC Fernzusteuern per Kamera und Rotation und Neigung automatisch zu bestimmet

Heftdrucker
https://www.youtube.com/watch?v=LqB32Myy-Hk
Ergebnis war dass mit https sehr langsam, ohne ging Sprachsteuerung im Browser nicht :-(… Kinder sind nicht drauf angesprungen.
Tinte ist eingetrocknet (Wassertropfen haben geholfen)
Konvertierung von Text in Bild und Übermittlung teils in Webassembly.

Weitere Ideen wie data firsct, derived contents … all platforms auch im Browser und weiteres … würde Ich gerne noch umsetzen und die Welt beglücken.

Sind aber auch Projekte entstanden wie eigene Rechnungsverwaltung und mehr.

2024 – 2044+

Warum 20 Jahre ? Die Ziele zu erreichen wird 10 Jahre dauern,
bis Communities übernehmen wirds nochmal 10 Jahre dauern
Ich bin ein bißchen müde immer neue Tools und Herausforderungen meistern zu

Tel.: +49 176/60032282 · E-Mail: marco-oweber@gmx.de, Im Tannhörnle 4/1 78052 Villingen-Schwenningen

müssen, und dennoch gegen Wände zu rennen. Egal wie groß der Hype im Internet dargestellt ist.
Beispiele

Nim: Return Types angeben kostet Zeit. Autor will nichts dran ändern.
Bidirectional Typing (wie z.B. in Haskell seid langem Standard ist wirds nicht geben)
Haxe: map(_ + 1) -> short lambdas -> Patch vom Autor abgelehnt
Rust -> Es gibt einen Grund warum es den Alloy Fork gibt (GC integriert).
Gleiches Problem wie bei Nim. Zusätzlich Shared Memory nutzen mit anderen Prozessen eventuell nicht so einfach. D.h. wenn man 100 GB im RAM und bischen Code schnell ändern will muss man alles neu laden.
TypeScript: Spart Zeit weil return Types oft von alleine erkannt werden.
Jedoch execve geht nicht so einfach oder macros fehlen (Ja bun aber .. IDE?)
Java/Koltlin/Dart/Haxe: OO lässt bestimmte Optimierungen nicht zu. GC ist für einige Anwendungen schlecht. Es gibt einen Grund warumt man bei Java Abhandlungen findet welche Art von GC bei welcher Anwendung mit welcher Konfiguration.
Siehe z.B. auch https://www.scylladb.com/scylladb-vs-cassandra/
Ähnlich die Verusche Java ohne Allocations zu schreiben.
Problem gibts bei Rust nicht.
Zig löst das Shared Memory Problem (Allocator immer übergeben), aber eventuell fehlen noch Libraries und ist noch sehr neu

Deswegen möchte Ich eine Entwicklungsumgebung schaffen, welche verschiedene Execution Contexts erlaubt, verschiedene Memory-Management Techniken je nachdem was wichtig für eine Aufgabe ist.
Und einen graduellen Übergang von kompiliert, interpretiert, jitted, dynamisch erlaubt und gleichzeitig SubTyping wie TypeScript integriert und bidirectionales Typing wie Haskell.

Execution Contexts Beispiele:
Webassembler > Browser > Server > DB
Verschiedene Schichten in Browser Extensions
Oder verschiedene Programmiersprachen
Transparente API Calls wie bei RakkasJS (runSSQ)

Shell Befehl Integration bzw Auflösung des Unterschieds Library / Exceutable

https://github.com/MarcWeber/gradualc

Während Ich Nixos Linux liebe, fehlen ein paar abstraktere schichten ähnlich

Tel.: +49 176/60032282 · E-Mail: marco-oweber@gmx.de, Im Tannhörnle 4/1 78052 Villingen-Schwenningen

requirements.txt (Python) welche es erlauben Abhängikeiten in einer Cross-Platform Cross Language Art und Weise definieren zu können, so dass als Nebenprodukt Entwicklungsumgebungen gebaut werden können und Pakete für Nixpkgs, chocolately, brew, conda, mamba, gentoo abgeleitet.

https://github.com/MarcWeber/it-glue

Die Probleme fallen mir immer wieder auf die Füße, deswegen fällt es mir schwer loszulassen. Und deswegen bin Ich der Meinung dass wir **gradual** Lösungen brauchen. Also so viel von einem Feature innerhalb einem Ökosystem wie sinnvoll. So wie TS,P< yright Typing Annotations erlauben. Oder Java allocation less geschrieben werden kann.

2023	Entwickeln von VSCode Plugins um die ähnlichen Projekte der vielen Kunden schneller zu erreichen lokal, VM und auf Server um Dateien abzugleichen oder SQL direkt im Editor auszuführen. Massive Zeitersparnis bei jeder kleinen Änderung von Configs und sonstigem Autoamtische Installationsscripte für PHP Systeme um Neukunden in 15min onboarden zu können.
	Entwicklen einer TS Bilbiothek für JSON Dokumente mit partieller oder vollständiger History mit 3 Speichformaten und Storage-Backends PG, MySQL, Files und IndexedDB (Browser) und Test-Suite. Leider wurde das Projekt wegen mangelndem Budget nie fertiggestellt.
	Beschäftigung mit OpenGL und Vulkan weil Ich bei Vesuvis Challenge mitmachen wollte.
2024	
	Email-Analyse von PNL-Fluggastendaten von Flughäfen für Webkos (PHP) mit Interface und Abgleich mit internen Passagierlisten. Idealer Workflow erreicht so dass auch wenn Emails in verschiedenen Formaten von verschiedenen Quellen kommen Konsistenz mit einem Blick geprüft werden kann
Fremdsprachen	Englisch: tägliche Kommunikation Französisch: Grundkenntnisse (nicht genutzt sehr rostig)
Führerschein	Klasse B,M,L (Auto + Motorrad)