




Date of birth: 29th November, 1990  
Telephone: (+420) 728 088 680  
E-mail: [kaja.valek@gmail.com](mailto:kaja.valek@gmail.com)  
Driver's Licence: B


---

**Education:** Masaryk University (Applied informatics, Informatics in public administration, graduated 2016), Faculty of Informatics, Brno  
Industrial polytechnics (Applied materials – leather, plastics and rubber), industrial design, graduated June 2011, Zlín

**Skills:**

**Development:**  **Python:** semi-advanced, active since 2012   
**JavaScript / TypeScript:** post-beginner-intermediate, active since 2019, TypeScript since 2023  
**Java:** beginner, basic understanding, some projects implemented in this language  
**GNU BASH:** intermediate  
**PHP:** Beginner, basic understanding, yet capable of implementing SW in OOP manner  
**C:** Basic understanding, yet capable of implementing SW.

**Operating Systems:** **Debian / Ubuntu:** intermediate, active use since 2006   
**Fedora / RedHat Enterprise Linux:** intermediate / semi-advanced, active since 2008, RedHat Internship  
For all of the systems above: Using command-line interpreter or SSH on daily basis


**Containerization:** **Docker / Podman:**  active use since 2020, intermediate, capable of writing own Dockerfile/Containerfile deployments  
**Kubernetes / Helm:** Beginner, incl Helm chart writing


**CI/CD:** **Gitlab CI/CD + GitHub Actions:** Practical experience, mostly with self-hosted runner.  
**Travis / Jenkins:** Brief experience

**Project tracking:** **Jira / Confluence**


---

**Experiences:**

**Prusa Development:** November 2022-April 2023  
Firmware developer (Python3  SH, Gitlab CI/CD – self-hosted runner). Development and tooling for automated 3D-printer testing.

**Kyndryl Client Centre / IBM Global Services Delivery Centre (GSDC) / AT&T Brno Czech Republic:**   
- **April 2022-September 2022:**  
Software Developer, internal project development and maintenance, mostly in PHP 5.x  
- **November 2020 – March 2022:**  
Software Developer, “Bridges Team”, brief maintenance of IBM CLOUD cluster running Kubernetes, TypeScript experience w/ implementing a simple micro-service solution.  
Employment change caused by employer (name change / division).  
- **August 2016 – September 2020:**  
Systems Deployment, AT&T, System installations (RHEL 7.x). Basic preparation for client and security check-ups.  
System management and preparation (Networking, data storage – SAN/Fibre, LVM).  
Employment change caused by acquisition by IBM GSDC, migration January 2020.

**Red Hat Czech Republic Brno:**





July 2015-June 2016  
Internship: Anaconda/Rhinstaller team, Junior Software Engineer, automated Blivet library  and Bachelor Thesis “Serialization of data storage configuration”.

**September 2014:** Network management and support (LAN), hosting professional multiplayer game tournament. Network infrastructure monitoring and correction, incl. Network infrastructure (eg. crimping, cable management).

## Projects:

- **“DevOps”:**
  - **Docker / Podman containers:**
    - Own-written Dockerfile / Containerfile solutions to either run solutions (for example Source Dedicated Server by Valve Software) incompatible with base systems
    - Development-assistance) dev containers – NodeJS, Java, TypeScript, .NET Core 5.x
  - **Self-hosted solutions:**
    - **Gitea** (Self-hosted repository solution to avoid dependency on 3<sup>rd</sup> party software like GitHub)
    - **Password manager / Vault**
    - Self-hosted frontend for “BarterBot” (Nginx)
  - **Kubernetes / Helm:**
    - Intention to learn how to operate Kubernetes as well as Helm
    - Self-study at home
    - Apply use for existing projects and studies
    - Completely-separated ecosystem self-hosted:
      - 1x Master
      - 2x Worker nodes
      - Self-hosted container registry using Docker
- **Development:**
  - **“BarterBot”:**
    - Formerly private project with focus on already-existing frontends (Telegram, Discord, ICQ)
    - Intention to stay active in development field
    - Implementations:
      - 1<sup>st</sup> – 10<sup>th</sup>: Python3, including ASYNC techniques (threading.Thread, asyncio)
      - 11<sup>th</sup> – 14<sup>th</sup>: NodeJS 14-16 / JavaScript, not utilizing basics without promises
      - 15<sup>th</sup>: TypeScript-based, major rewrite
    - Custom front-end using Vue3.js (TypeScript) as a own solution without dependency on 3<sup>rd</sup> party-projects
    - This project was and is also used by other clients as well, experience gained here was reused in next implementations
    - Multiple backends in various programming languages
      - SourcePawn
      - Java + Apache Maven (Oracle-based JDK / JRE, self-designed Docker container).
      - Connected via TCP/IP-based socket, data exchange in JSON
      - HTTPS Express-based (TypeScript / NodeJS) REST API (mainly for data exchange with Vue.JS frontend), Let’s Encrypt.
  - **AT&T: Management Portal:**
    - During employment in AT&T, 2019
    - Including proposed software architecture:
      - Backend: Python3 (LDAP + PostgreSQL interactions)
      - Frontend: PHP 7.x
      - Both endpoints connected via TCP/IP socket-based communication in serialized in JSON.
    - Actively using AT&T’s proprietary login solution.
  - **“screxec / scrstart” (“ScreenTools”):**
    - Developed own solution to mimic terminal multiplexer (SuperPutty), but fully BASH and text-only based
    - Implemented this solution to assist my daily tasks in AT&T.
  - **“FlaskRestfulPython”**
    - Very simple website with REST API with intention to learn Flask framework written in Python
    - Consists of simple Web user interface as well as REST API
    - Purpose was to share files with both via HTML-based UI as well as REST API’s methods (GET, POST, PUT, DELETE)
    - Custom-defined Docker container, service is executed as a self-contained application within the container
  - **“AnsibleTest”**
    - Adopt Ansible and methods how to work with Ansible
    - Self-study purpose

- **“Selenium-Playwright”**
  - Formerly projects for **Selenium** framework, but also containing **Playwright**.
  - Understanding of both frameworks and their documentations.
  - Intended to be used as a self-test measure for BarterBot WebClient framework
  - **Programming languages used for development:**
    - Java (Apache Maven + JUnit)
    - Python (VirtualEnv)
  - Secondary experiences mainly with Docker container to create self-developed environment for Java development:
    - Based on Ubuntu 22.04
    - openjdk
    - Xorg-Xserver (vncserver)
  
- **“Django” (GraphQL):**
  - Mini-project, exclusively for Django Framework (Python).
  - GraphQL branch: self-study and full CRUD implementation for GraphQL API.
  - Self-study.
  
- **“MongoDBPlayground”**
  - Combined project of both Development and DevOps
  - Full deployment of a MongoDB inside a self-hosted Kubernetes cluster.
  - Two implementations:
    - Python (VirtualEnv, PyTest)
    - Java (Gradle + Junit)
  
- **“Software-controlled-Hardware” / Microcontrollers**
  - **Raspberry Pi semi-automated gardening**
    - Python-based, simple script that accepts arguments via CLI (automation via CRON)
      - Time, GPIO pin
    - Relays connected to Raspberry Pi’s GPIO headers to control externally-powered pumps to automatically irrigate home-planted crops.
  
  - **Relay-power-oner**
    - Arduino-based (C Language), communication via additional serial line with remote master computer.
    - Set of 6 independent relays connected to separate Arduino R3 pinouts
  
  - **Rpi-Lighting / WS2812b**
    - Formerly Raspberry Pi Zero W based project, code written in Python
    - Connected to an Android application via Bluetooth protocol, simple string-based protocol to control color and power of set of 8 RGB LEDs.
    - Current version based on ESP-32 based hardware, scaled up to ~250 LEDs (75W 5V PSU)
      - Same Bluetooth-based protocol.
  
  - **“External WOL box”**
    - Arduino-based
    - Ethernet + SD Card shield
    - Firmware / ROM written in C language
    - Purpose to automatically wake-up a computer via LAN, with MAC addresses stored inside a text file stored on a SD card.
  
- **General:**
  - **VirtualBox:**
    - Active use.
    - Mainly for development inside a Debian machine
    - “Framework” to run containers for development (Java, TypeScript, etc).
  
  - **KVM - Libvirt / virsh / virt-manager:** Hobby experience, ran a Windows 10 OS fully virtualised on a IOMMU-based machine (Debian Host, Windows 10 guest, dedicated 10/12 of CPU cores).
    - Includes hardware pass-through (namely nVidia GPU).

- **LXC**
  - Simple gaming server.
  - Munin monitoring
- **Visual Studio Code:**
  - daily basis
  - Multi-profile development:
    - Python
    - Java
    - TypeScript / JavaScript
  - Development heavily using remote means:
    - Docker/Podman containers
    - via SSH
- **MS Windows:** Capable, daily-use
- **Windows Server:** Basic experience, mostly with 2019 / 2008
- **3D Printing / 3D-Design**
  - **Upcycled, home-made PC displays** 
    - Using broken laptop's LCD Panels.
    - Measurements, 3D design and printing on home capacities
  - **Gardening:**
    - Set of 3D-printed frames and components from PETG and TPU materials as part of automated gardening via Raspberry Pi computer
    - Two versions:
      - Multi-pump
      - Single-pump (Active) 
  - **"The Totem"**
    - 3D-engineered and printed case for **"Rpi-Lighting / WS2812b"** project.
  - **Various monitor stands, VESA adaptors**
    - Accompanied as a secondary mini-project for Upcycled PC displays.
    - Adaptor for VESA-proprietary DELL's rear monitor attachment
  - **Elevation Table**
    - Big-sized, multi-part project for elevating 2D planar scanner above a conventional printer.
    - Custom 3D-design engineering to address weight to 3D-printed material characteristics (tensile strength)
  - **Various laptop stands**
    - Up to 5-6 kilograms of weight or for lighter laptops.
    - Purpose to create additional space
  - **Magnifier stand**
    - Special construction used as a cell phone holder to act as a magnifier
  - **Maintenance-related:**
    - Construction and maintenance of a 3D-printer (bearings, custom casings, cable management) 
    - Design and construction of various 3D-printers out of "scrap", leftover material and structural parts.
  - **Mentoring / assistance**
    - Utilizing experiences to help my clients and external 3D-printer owners 
      - How to use 3Dprinter
      - How to design custom parts
      - How to correctly set up a slicing software
    - 3Dprinter maintenance on client's demand
  - **"External WOL box"**
    - Custom casing for mini project with same name (Arduino)