Curriculum Vitae

Andrii Nemchenko
telegram: @tellnobody1
email: a.nemchenko@icloud.com
location: Kyiv, Ukraine

February 3, 2019

1 Experience

1.1 Web Developer (Scala, PureScript)

6 years (Apr 2013 – Present)
Playtech

Description Developing and evolving Web Platform for licensees to create portals with player's account management, games hub, content-management system (CMS) and integration with 3rd-party services and data providers (AS).

1.2 Frontend Developer (Java)

1 year (Apr 2012 – Mar 2013)
Playtech

Description Developing cross-platform mobile games (HTML5 Canvas, iOS, Android) using modified Google PlayN framework to meet performance expectations for players. Use CSS 3 for native 3D HTML games. Creating a common framework for slot games to eliminate code duplication and deliver large number of titles in tight schedule.

1.3 Web Developer (PHP)

2 years (from Apr 2010 – Mar 2012)

Description Writing not so big websites and also e-commerce, survey analytics platform, social networks for local companies, corporations and startups. Diving into PHP, MySQL, using some CSS, JavaScript, looking at Photoshop.

2 Education

2.1 Kyiv National Taras Shevchenko University

Master of Science (2006 – 2012)
Applied mathematics

3 Certification

3.1 Functional Programming Principles in Scala

EPFL (École polytechnique fédérale de Lausanne)
Dec 2013

4 Projects

4.1 KVS

Jul 2014 – Present
Playtech

Stack Scala, Akka (cluster, cluster-sharding), LevelDB (via jnr-ffi)

Description Closed-sourced abstract Scala storage framework with high-level API for handling linked lists of polymorphic data (feeds).
Designed with various backends in mind and to work in pure JVM environment. Implementation based on top of KAI (implementation of Amazon DynamoDB in Erlang) port with modification to use akka-cluster infrastructure.
Currently main backend is LevelDB to support embedded setup alongside application. Feed API (add/entries/remove) is built on top of Key-Value API (put/get/delete).
KVS is highly available distributed (AP) strong eventual consistent (SEC) and sequentially consistent (via cluster sharding) storage. It is used for data from sport and games events. In some configurations used as distributed network file system. Also can be a generic storage for application.

4.2 CMS

Apr 2013 – Present
Playtech

Stack Scala, Akka (cluster-sharding, http-core), PureScript, Scala.js, Scalaz (core, effect), Argonaut, Scodec, Protobuf

Description Content management system with one cluster setup to serve dozen licensees each with up to 10 websites. CMS is all about storing data and working with UI. Data is stored to distributed KVS and UI is built with pure functional strongly typed language (PureScript) which produces robust and fail-safe UI. User files are handled by distributed filesystem and meta data is saved to KVS.

4.3 AS

Jul 2014 – Feb 2017
Playtech

Stack Scala, Akka (actor, cluster, stream, http-core), Cats (effect), Scalaz (core), Argonaut, Pickling

Description Application server with streaming idea in its core. The integration layer for services providers which unifies the different APIs and respect the providers limitations guarding their services from unexpected usage. Unified services structure with akka-stream based IO layer and KVS distributed storage engine. Also plays as the service provider for itself to provide event streaming for sports betting and additional data store interfaces for the client application.

4.4 Monitoring

Feb 2015 – Present
Playtech

Stack Scala, Akka (stream), PureScript, React.js, UDP, JMX

Description Monitoring application receives the metrics, stores and display them in the dashboard. With some basic analyse and triggers the monitor can notify the interested parties in the specific events (application crash, node down, specific user event occurs, etc). Visualization and performance reporting.

4.5 Documentation

Stack Scala, Akka (http-core), LuaLaTeX, Hevea

Description Documenting system and server for the applications. LaTeX based tools for generating the project documentation in HTML and PDF format.

4.6 Open API

Stack Java, OSGi

Description The platform provides the components to build the services which will be available to clients through unified Open API. The Framework is the set of applications which are bundled together as an OSGi features and OSGi bundles.

5 Interests

Blockchain, Compilers, Navigation
Idris, Haskell, PureScript, Erlang, Rust, Swift, Kotlin
HoTT, CT, Coq, Quantum computing