Curriculum Vitae

Andrii Nemchenko
telegram: @tellnobody1
email: andriinemchenko@gmail.com
location: Kyiv, Ukraine

September 4, 2020

1 Experience

2020 Lead Developer at Playtech plc[1]

2017–2019 Software Architect

2014–2016 Senior Scala Developer

2013 Functional Programming Principles in Scala at EPFL[2]

2013–2014 Senior Java Developer

2012–2013 Middle Java Developer

2010–2012 Full-Stack Web Developer

2010–2011 English level B2 (CEFR) at FSCFL[3]

2006–2012 Applied mathematics (M.S.) at TSNUK[4]

2 Public profiles

GitHub: @zero-deps and @flyingw
Stack Overflow: Andrii Nemchenko
HackerRank: a_nemchenko

[1] https://www.londonstockexchange.com/stock/PTEC/playtech-plc
[2] https://www.epfl.ch/en/
[3] https://fcourses.com.ua
[4] https://www.univ.kiev.ua/en/

3 Projects

3.1 Key-value storage

GitHub: zero-deps/kvs

Stack Scala, Akka Cluster, LevelDB

Description 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.

3.2 Web Platform

Stack Scala, ZIO, Akka Cluster, PureScript, Protocol Buffers

Description Developing and evolving Web Platform for licensees to create portals with player's account management, games hub, content-management system and integration with 3rd-party services and data providers.
It is deployed as one cluster for dozen of licensees with dozens websites. Data is stored into 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.

3.3 Application Server

Stack Scala, Akka Cluster, Akka Streams

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.

3.4 Monitoring

Stack Scala, Akka Streams, PureScript

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.

3.5 Documentation

Stack Scala, LaTeX, HeVeA

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