

## EDUCATION

---

- **Moscow Aviation Institute** Moscow  
*Master in computer science and mathematics* *Sep. 1998 – Feb. 2005*

## EMPLOYMENT

---

- **Nordea Bank** Copenhagen, Denmark  
*Expert Developer, Chief IT Developer (since 2021), Team Lead* *Feb 20015 - Present*

### *Achievements:*

- Designed and led implementation of several key systems.
- Worked as a “crisis” tech lead to stabilize problematic projects.
- Worked across teams to promote best programming practices.

*Tech stack:* Kotlin, Java, Typescript, C-sharp, Kafka, GRPC, Protobuf, Avro, Docker, Cassandra, Redis, Mongo, Oracle, ActivePivot, React, OpenAPI

- **Deutsche Bank** Moscow, Russia  
*Senior Software Engineer (AVP)* *Apr 2012 - Nov 2014*

### *Achievements:*

- Designed and implemented backend of risk management system.

*Tech stack:* Java, Oracle, Oracle Coherence

- **Luxoft** Moscow, Russia  
*Senior Software Engineer* *March 2011 - March 2012*

### *Achievements:*

- Development of a low latency price distribution system.

*Tech stack:* Java, NIO, Multicast

- **Deutsche Bank** Moscow, Russia  
*Senior Software Engineer* *Apr 2005 - Nov 2010*

### *Achievements:*

- Designed and implemented number of critical pricing and trading systems.

*Tech stack:* Java, Sockets, JMS, Oracle

## SIDE ACTIVITIES

---

- Strategy back-tester for algo trading in kotlin, java, python.
- Telegram bot for technical analyses (<https://web.telegram.org/z/1072760314>).
- AI research for trading strategies with python, pandas, numpy, scikit-learn libs.

## SKILLS

---

- **Languages:** Kotlin, Java, C-sharp, Typescript, Python, SQL
- **Technologies:** Kafka, Docker, Mongo, gRPC, React, Avro, Protobuf, Cassandra
- **Other:** gradle, bash, unix, computer science, git, ML knowledge, pandas, scikit-learn, numpy

## MY DEVELOPMENT PRACTICES

---

- Complexity and technical debt management is in the top of my objectives while designing and implementing project. Improper management often leads to an exponential complexity growth to the system functionality.
- While leading a team I try to avoid redundant people sidetracking and context switching to micro tasks as this is very expensive.
- Cross team communication can be greatly improved by extensive usage of systems contracts/APIs, as people are "slow" computers are "fast".
- I prefer functional approach over OOP which is often overused.