* [Facing the Unplanned Migration of Serverless Applications: A Study on Portability Problems, Solutions, and Dead Ends](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/unplanned%20migration%20of%20serverless.pdf)
  + Manual migration across 3 cloud provider: AWS, Azure, IBM Cloud
* [ToLambda--Automatic Path to Serverless Architectures](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/automatic_path_to_serverless_architecture.pdf)
  + Automatic migration from Java monolith -> AWS Lambda NodeJS
  + It makes language (from Java to JS) and structural transformation.
* [Lessons Learned from Migrating Complex Stateful Applications onto Serverless Platforms](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/lessons_learned_from_migration_complex_stateful_app.pdf)
  + Manual migration of 4 complex and stateful microservice applications to Apache OpenWhisk
  + It highlights several patterns and guidelines that facilitate the migration with minimal code changes and practical performance considerations.
  + Future direction: a tool that helps developers migrate their legacy microservice applications to serverless or that automates this migration.
* [Migrating from Monolithic to Serverless: A FinTech Case Study](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/migration_from_monolith_to_serverless_finTech.pdf)
  + Manual migration from a monolithic architecture to a serverless architecture
  + They identified the barriers and challenges in migration process.
* [The SPEC Cloud Group’s Research Vision on FaaS and Serverless Architectures](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/*The_spec_cloud_group_s_research_vision_on_faas_and_serverless_architectures*.pdf)
  + It presents research directions and opportunities.
  + Challenges and opportunities: investigate how to optimize the migration process, and to what extend it is possible to automate the extraction of functions from legacy systems to cloud functions.
* [Survey on serverless computing](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/*The_spec_cloud_group_s_research_vision_on_faas_and_serverless_architectures*.pdf)
  + They examined 275 research papers and presented stated of the art contributions, as well further research possibilities.
  + Migration of monolithic applications to serverless computing (RQ9)
  + Future direction of research: finding optimal automatic migration solutions for existing legacy systems, research on tools for checking whether a legacy system will fit the serverless paradigm.
* [The Journey to Serverless Migration: An Empirical Analysis of Intentions, Strategies, and Challenges](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/journey-to-serverless-migration.pdf)
  + They investigated the migration processes of 11 systems across diverse domains by conducting 15 in-depth interviews with professionals from 11 organizations.
  + Technical challenges: Identifying the use cases best fit for the serverless architecture and splitting the system.
  + most systems were migrated gradually using strangler patterns to serverless.
  + legacy monolithic systems were decomposed into microservices and migrated to serverless by applying domain driven design approach, decomposition by database.
* [A Systematic Mapping Study on Engineering Function-as-a-Service Platforms and Tools](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/systematic%20mapping%20study%20on%20engineering%20FaaS.pdf)
  + They collected and synthesized data provided from 62 publications and identified research gaps for future research.
  + Research category: migration of legacy application to FaaS service model. Automating the migration process are interesting topic in this area.
* [Partial Migration for Re-architecting a Cloud Native Monolithic Application into Microservices and FaaS](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/PartialMigrationforre-architectingacloudnativeMonolithicApplicationintoMicroservicesandFaaS_ICICCT_2020_170.pdf)
  + They applied unsupervised learning algorithm on web server access logs of one complete application execution cycle to map components that best suits monolith-microservices-FaaS service patterns.
* [A Rule-based System for Automated Generation of Serverless-Microservices Architecture](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/rule%20based%20system%20for%20automated%20generation%20of%20serverless.pdf)
  + It proposes TheArchitect tool, a rule-based system, to automate the serverless-microservices based high-level architecture generation process.
* [Rise of the Planet of Serverless Computing: A Systematic Review](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/*rise%20of%20the%20plantet%20of%20serverless%20computing.pdf)
  + This paper provides a comprehensive literature review of 164 analyzed papers
  + Research direction: Application migration. They mentioned that for manual conversion approaches major studies have been specific to a certain class of applications (e.g web applications, AI-related applications). Regarding automated conversion approaches are still two problems to be addressed. First, how to automate conversions is a critical question, since a mature approach is lacking. Second, providing a generic application migration approach is challenging for different application types.
  + Opportunities for researchers: Generalizability of application conversion approaches. Existing application conversion approaches have targeted only a few specific applications, such as AI applications, Web applications, and Java applications. There are not yet generic conversion tools for any application.
* [A Framework for Bridging the Gap between Monolithic and Serverless Programming](https://github.com/Nimrod-F/research-cold-start/blob/main/articles/migration/16-depalma-framework-monolithic-serverless.pdf)
  + They presented a framework which transform JS monolith -> JS separate serverless functions based on annotations, basically it’s a transpiler.