```clojure
(ns my.resume
  (:require
    [clj-time.core :as t]
    [my.util :refer
     [time-range
      locations]]))

(def contact-info
  {:first-name "Anton" :last-name "Shkalev"
   :birtdate (t/date-time 1989 3 9)
   :email "anton.shkalev@gmail.com"
   :phone-number "+7-926-106-1620"
   :location (locations :Moscow)})

(def experience
  [{:job-title "CTO, co-founder" :company-name "GetShopApp"
    :time-range (time-range [2013 4] nil)
    :projects
    [{:title "Getshopapp.com - m-commerce platform"
      :roles ["System architect" "Senior back-end engineer" "DBA" "DevOps engineer"]
      :description "We were creating an m-commerce platform for existing e-shops. \
                    Our main goal was to provide exceptional shopping experience \
                    for mobile visitors of the online shop. We created native \
                    apps and mobile-friendly versions of the desktop sites with \
                    deep integration to shop's existing e-commerce platforms."
      :responsibilities
      ["Designing a robust and scalable back-end architecture"
       "Creating a JSON API for apps, admin panel, external integration modules"
       "Creating an automated app publishing system for Google Play and AppStore"
       "Working with contractors (developers of intergration modules)"
       "Working with a team of designers on front-ends sketches"
       "Hiring developers (incl. Skype and on-site interviews)"
       "Maintaining company servers and rolling out updates"]}]}

   {:job-title "Software engineer" :company-name "Intel Russia"
    :time-range (time-range [2008 8] [2013 4])
    :projects
    [{:title "Research and evaluation of a new CPU architecture"
      :roles ["Component owner" "Developer" "Analyst"]
      :description "NDA"
      :responsibilities
      ["Developing and maintaining components of a CPU performance simulator"
       "Implementing and evaluating new optimisation techniques"
       "Fixing bugs" "Refactoring"]
      :awards
      [{:title "For timely delivery of high quality report to <project-name> \
                reviewers"
        :level :cross-business-group :scope :personal}
       {:title "For highly performant work, leading to achieving key result \
                ahead of schedule"
        :level :department :scope :team}
       {:title "Teamwork Award level 2" :level :co-worker :scope :personal}]}

     {:title "Compiler for Itanium architecture"
      :roles ["Component owner" "Developer"]
      :description "Maintenance of codegen of Intel Compiler for Itanium"
      :responsibilities ["Maintaining a component of a huge SW project"]
      :awards
      [{:title "For surpassing initial expectation of maintaining bugs backlog \
                at current level by bringing it down to zero"
        :level :department :scope :team}]}]}])

(def expertise
  {:back-end
   {:description "Developing back-ends for challeging projects is a pleasure \
                  for me. I love Clojure for it's elegance and flexibility. I \
                  also use Python for scripting and small subprojects. I know \
                  SQL (My- and Postgre-) and prefer it (with a touch of query \
                  generators, like korma) to ORMs. Mostly writing HTTP APIs - \
                  so I know this protocol to a great extent."
    :buzzwords #{:rest :crud :json :http :api}}}
```

```clojure
    :system-design
    {:description "I have experience in design of both monolithic and service-based \
                   systems, leaning towards later. Aware of pros and cons of \
                   stateful/stateless services. Know core principles of high-load \
                   projects design. Believe in traditinal SQL datastores. \
                   For Getshopapp I made a clustered SOA back-end with configurable \
                   run-time distribution of roles and fault-tolerance (was standing \
                   on the shoulders of giants, though)."
     :buzzwords #{:microservices :cloud :soa :ha :cap-theorem :load-balancing}}
    :dev-ops
    {:description "I can maintain a small/medium fleet of servers on AWS. \
                   A big fan of Linux and it's philosophy (should I call it \
                   'The Unix Way'?). I use Arch and FreeBSD at home, but can work \
                   with Debian and Ubuntu as well. \
                   Getting in touch with CoreOS and familiar with Docker, wrote \
                   a simple utility in Python (nothing special) to cover the rough \
                   edges of not yet mature technology. Very excited about it, though."
     :buzzwords #{:aws :cloud :ci :docker :linux}}
    :dba
    {:description "I have some experience in creation of DB schemas. Extensively \
                   use foreign keys and constraints. Know how to analyze query \
                   performance, about the importance of indices (and the correct \
                   order of columns in it), about sharding and hierarchical tables, \
                   normalization and denormalization."}
    :ios-development
    {:description "Have some experience in iOS development as well. I know ObjC at \
                   a reasonable level, know how to distribute tasks between threads \
                   and queues, how to use ARC and /not/ leak memory with blocks. \
                   Not so good at UI tasks, though."}})

(def skills
  "A short list of *some* relevant skills I have, ordered by experience level"
  {:languages [:clojure :python :bash :objective-c :c :c++ :javascript]
   :platforms [:linux :aws :jboss :docker :core-os]
   :databases [:mysql :postgresql]
   :tools [:git :vim :xcode :mercurial :svn]
   :org [:jira :redmine :bugzilla :agile]})

(def education
  [{:type :higher
    :institution {:title "Lomonosov Moscow State University"
                  :location (locations :Moscow)
                  :type :university}
    :time-range (time-range [2006 9] [2011 6])
    :field "Applied Mathematics and Computer Science"
    :degree :master}
   {:type :secondary
    :institution {:title "Advanced Education and Science Center, MSU"
                  :location (locations :Moscow)
                  :type :school}
    :field "Computer Science"}])

;;; cut here ;;;

(ns my.util
  (:require [clj-time.core :as t]))

(def locations
  "A map of locations, used in this document. With ISO-3166-2 codes."
  {:Moscow {:subdivision-name "Moskva"
            :subdivision-category :autonomous-city
            :code "RU-MOW"}})

(defn time-range
  "Creates a vector of two Joda DateTime objects, representing a time range"
  ([start]
   (time-range start nil))
  ([start end]
   {:pre [(not-empty start)]}
   (mapv #(or (some->> (not-empty %)
                       (apply t/date-time))
              (t/now))
         [start end])))
```