The Travel Guide to Software Systems

Patrick Kua (@patkua)
October 2017
Digital Solution to Banking

- Full European banking license
- First fully mobile bank account
- One-touch access to all financial products
500.000+ users in 17 countries

With more than €5.5bn transaction volume
Full Bank account
Built from the ground up for mobile

Seamless integration
Insurances, Savings, Credit
Overdraft
Flexibility in seconds

Credit
Up to 25k Euro payout in real-time
N26 Invest
Investing with just a few taps

N26 Savings
Save with partner banks across Europe

Plan Projection

Select Savings Offer

Duration in months: 6, 12, 24, 36, 48, 60

12 Months
- First Save Euro: 0.85%
- Inbank AS: 0.80%
- Granka Bank AG: 0.50%

24 Months
- First Save Euro: 1.00%
- Inbank AS: 1.00%

Risk Disclaimer

€2000
One-time amount

€200
Monthly amount
Consistent strong recognition by Apple and Google

Featured as “Editor’s Choice”
Featured at Apple WWDC
Several promotions through Apple

Featured in “Best Apps in 2016”
Featuring in the Play Store with major releases
Users love the N26 experience

5,685 Reviews
I am building the technology group behind the mobile bank designed for the digital age.

We are looking for people to join us on that journey: [https://n26.com/jobs/](https://n26.com/jobs/) (Berlin)

#leader #coach #architect #developer
#life-long-learner #speaker #author
CTO of N26

[Book covers for "Building Evolutionary Architectures" by Neal Ford, Rebecca Parsons & Patrick Kua, "Talking with Tech Leads: From Novices to Practitioners" by Patrick Kua, and "The Retrospective Handbook: A guide for agile teams" by Patrick Kua]

[Links: thekua.io/evolarch, thekua.io/twtl, thekua.io/retrobook]
Our itinerary
Our itinerary
Why we might consider using a travel guide
You find yourself here
You find yourself here

Where are you?

What dangers lie ahead of you?

How do you communicate?

How do you achieve your goal?
static inline void
adfs_objdir(struct adfs_c_entry *de, struct object_info *obj)
{
    adfs_writeval(de->direaddscadd, 3, obj->file_id);
    adfs_writeval(de->direaddload, 1, obj->loadaddr);
    adfs_writeval(de->direaddexe, 1, obj->execaddr);
    adfs_writeval(de->direaddlen, 1, obj->size);
    de->newdirsets = obj->value;
}

/*
 * get a directory entry.  Note that the caller is responsible
 * for holding the relevant locks.
 */
static int
__adfs_dir_get(struct adfs_dir *dir, int pos, struct object_info *obj)
{
    struct super_block *sb = dir->eb;
    struct adfs_c_entry de;
    int thsize, offset, offset;
    buffer = pos >> ad->a_blocksze_bits;
    if (buffer > dir->a_buffers)
        return -EINVAL;
    offset = pos & (ad->a_blocksze - 1);
    thsize = sb->a_blocksze - offset;
    if (thsize > 26)
        thsize = 26;
    memcpy(sb, dir->a_buffer + offset, thsize);
    if (thsize < 26)
        memcpy((char *)de, thsize, dir->a_buffer[buffer - 1] + offset);
    if (!de.diren��name[0])
        return -ENOSK;
    adfs_objdir(dir, obj, de);

    return 0;
}
if (!oe->dir->name[0])
    return -ENOPRT;

ads_cdir2obj(dir, obj, sde);

return 0;

static int
_ads_cdir2obj(struct ads_dir *dir, int pos, struct object_info *obj)
{
    struct super_block *sb = dir->sb;
    struct ads_cdir_tree *ctree;
    int tthisize, buffer, offset;

    buffer = pos >> sb->sb_blocks->a_blocks;

    if (buffer > dir->nrr_buffers)
        return -EINVAL;

    offset = pos & (sb->sb_blocks->a_blocks - 1);
    tthisize = sb->sb_blocks->a_blocks - offset;
    if (tthisize > 26)
        tthisize = 26;

    /*
     * Get the entry in total
     */
    memcpy(sb, dir->bh.buffer->b.data + offset, tthisize);
    if (tthisize == 26)
        memcpy((char *)&sde) - tthisize, dir->bh.buffer[1] - 1->b_data,
        26 = tthisize);

    /*
     * Update the entry in local
     */
    adn_obj_dir(dir, sde, obj);

    /*
     * Put the new entry back
     */
    memcpy(dir->bh.buffer[0]->b.data - offset, sde, tthisize);
    if (tthisize == 26)
        memcpy(dir->bh.buffer[1] - 1->b_data, ((char *)&sde) + tthisize,
                26 - tthisize);
if (ioe.dirname[0])
    return -ENOTRMT;

adfs_dir2obj(dir, obj, sde);

return 0;

}  

static int
__adfs_dir_put(struct adfs_dir *dir, int pos, struct object_info *obj)
{
    struct super_block *sb = dir->sb;
    struct adfs_dientry de;
    int thisize, buffer, offset;

    buffer = pos >> sb->s_blocks_per_bits;

    if (buffer > dir->nr_buffers)
        return -ENOSPC;

    offset = pos & (sb->s_blocksize - 1);
    thisize = sb->s_blocksize - offset;

    if (thisize > 26)
        thisize = 26;

    /*
     * Get the entry in total
     *
     * memmove(sde, dir->bh.buffer + b.data + offset, thisize);
     * if (thisize > 26)
     *     memmove((char *)sde + thisize, dir->bh.buffer + 1)->b.data,
     *              26 - thisize);
     */

    /* update it */

    adfs_obj2dir(sde, obj);

    /*
     * put the new entry sde
     */

    memmove(dir->bh.buffer)->b_data - offset, sde, thisize);

    if (thisize > 26)
        memmove(dir->bh.buffer + 1)->b_data, ((char *)sde + thisize,
                          26 - thisize);
The travel guide

You find yourself here
Why travel guides?
Why we should care

- People join/leave
- Scaling organisations
- Leaving a legacy
- Preventing (unwanted) detective work
Alternatives?

- Nothing
- Viking tales
- Telephone game
Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
Manifesto for Agile Software Development

Working software over comprehensive documentation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
Our itinerary

What we need
Context, context, context
Stories of why the system emerged
(Hard) lessons learned
Significant decisions (ADRs)
Common traps, pitfalls, known tech debt
Domain facts
Domain vocabulary
“Page of Acronyms”
Personas - who uses the system
Common system use cases
Travel Guide

- History
- Useful facts
Culture

Coding standards
Code review/pair programming
Conventions in the codebase
Common programming patterns
Development process
Travel Guide

- History
- Useful facts
- Culture
Maps of the system
Simon Brown’s C4 Model (Context, Containers, Components, Classes)
Sequence diagrams, flow charts, CRCs
What are the “landmarks” of your codebase?
Some areas are more important than others
Most commonly travelled parts of the system
Fun facts about the system
Interesting areas of the system
Places where innovation/creative solutions exist
Our itinerary
How we make the most of our time
Starting point
Size
Keeping It alive
What’s valuable for others?
FAQs
Help your future self
Size

\[ f(\text{History, Population, Degree of change, Speed of change, Domain complexity}) \]
\[ f(\text{Size, History, Population, Degree of change, Speed of change, Domain complexity}) \]

Size

Smaller

Larger
Value Over Size

“I would have written a shorter letter, but I did not have the time”
- Blaise Pascal
Software systems don’t stay stable
Software systems don’t stay stable

Keeping it alive

New York City

10th edition (2016)


Neither do cities!
Keeping it alive

Ideate -> Test -> Review

Pareto Principle

Generate -> Ideate
Our itinerary

Compare our experiences with others
http://www.seleniumhq.org/

Started in 2004
Web automation tool
Combined with WebDriver
One of the most widely used web testing tools
Selenium Documentation

Contents:

- Note to the Reader - *Docs Being Revised for Selenium 2.0!*
- **Introduction**
  - Test Automation for Web Applications
  - To Automate or Not to Automate?
  - Introducing Selenium
  - Brief History of The Selenium Project
    - Selenium’s Tool Suite
    - Choosing Your Selenium Tool
    - Supported Browsers and Platforms
    - Flexibility and Extensibility
    - What’s in this Book?
    - The Documentation Team–Authors Past and Present
- Selenium-IDE
http://www.eclipse.org

A popular IDE

Plugin architecture supporting many languages, tools

Basis of many other products
http://www.eclipse.org
http://getfirefox.com

Alternative open-source browser
Focus on open standards
Huge community
Documentation topics

Getting Started
A step-by-step beginner's guide to getting involved with Mozilla.

For new Mozilla developers
A directory of articles which are particularly helpful for new Mozilla developers.

Working with Mozilla Source Code
A code overview, how to get the code, and the coding style guide.

Build Instructions
How to build Firefox, Thunderbird, SeaMonkey, or other Mozilla applications.

Editor Configuration
Tips on setting up your favorite IDE or text editor to work with Mozilla projects.

Development process overview
An overview of the entire Mozilla development process.

Managing multiple profiles
When working with prerelease versions of Firefox, it's often helpful to have multiple Firefox profiles, such as one for each channel, or for different kinds of testing.

Automated Testing
How to run Mozilla's automated tests, and how to write new tests.

How to submit a patch
After getting your patch written, you need to get it checked into the tree. This article explains the review process and how to get your patch approved.

Tools

Bugzilla
The Bugzilla database used to track issues for Mozilla projects.

DXR
Next generation of searching Mozilla's source code. In active development.

SearchFox
Another option for Mozilla code searching. Indexes JS as well as C++, includes blame capabilities. In active development.

Mercurial
The distributed version-control system used to manage Mozilla's source code.

Mozilla build VM
A VirtualBox compatible virtual machine configured with all the software needed to build and work on Firefox.

TaskCluster
TaskCluster is the task execution framework that supports Mozilla's continuous integration and release processes.

Treeherder
Treeherder shows the status of the tree (whether or not it currently builds successfully). Check this before checking in and out, to be sure you're working with a working tree.

Perfherder
Perfherder is used to aggregate the results of automated performance tests against the tree.
Our itinerary

Paying it forward
Working software over comprehensive documentation

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
Prepare for the trip
Prepare for the trip
Build your own guide
Prepare for the trip
Build your own guide
To help others enjoy the ride
Thank you

Psst. We’re hiring in Berlin
https://n26.com/jobs

PATRICK KUA, CTO
@n26, @patkua