

Erlang in E-Commerce and Banking

or

Painless Payment Processing

by

Erik Stenman

CTO, Kreditor Europe AB

KREDITOR

Säljande betalningslösningar
Creative Payment Solutions

Kreditor

- The business model:
 - Bring trust to Internet shopping.
 - Bring old style billing into the new IT-economy.
- Brief background:
 - Founded in **February 2005**.
 - With < 60,000 € in venture capital.
 - Live system in **March 2005**.
- The company vision: “Be the coolest company in Sweden.”

KREDITOR

Success So Far

- More than **2000 web-shops** are using our system, among them several of Sweden's largest.
- Expanded from 3 to **55 employees** without using VC.
(And ten more are joining the tech department this summer/fall.)
- **Making profit** since year 2, while expanding.
- **Profit 2007: just under 500.000 €.**
- **Monthly turnover up 40%** January-April.
- Introduced more than eight **major** new services since start.
- **Daily** upgrades to the system, **without any downtime.**

KREDITOR

The Problem

- Internet shopping is a question of **trust**.
 - The **shop** has to **trust** the **customer** to get paid.
 - The **customer** has to **trust** the **shop** to send the stuff.
- Many (Swedish) **customers** are **uncomfortable** using credit card over the Internet.
- Many banks are actually **worried about the security** of Internet **shops** handling credit card information.
- Also, doing a *partial return* when using credit card is a **hassle**, both for the **customer** and the **shop**.

KREDITOR

The Solution

- Bring in *a trusted party*, i.e., **KREDITOR**.
- Send an invoice to the **customer**.
- The **customer** pays after receiving the goods and takes no risk. The **customer** does **not** have to trust anyone.
- The **shop** is guaranteed (by contract) to get money from **KREDITOR**. The **shop** only have to trust **KREDITOR** with whom they have a written contract.

KREDITOR

Advantages

- The **customer** gets credit.
- The **customer** can pay using known methods.
- Returning goods is easy.
- **Better fraud detection.**
- **Advanced credit assessments.**
- Easy to add similar features like *pre-pay*, *instalment plans* and *subscriptions*.

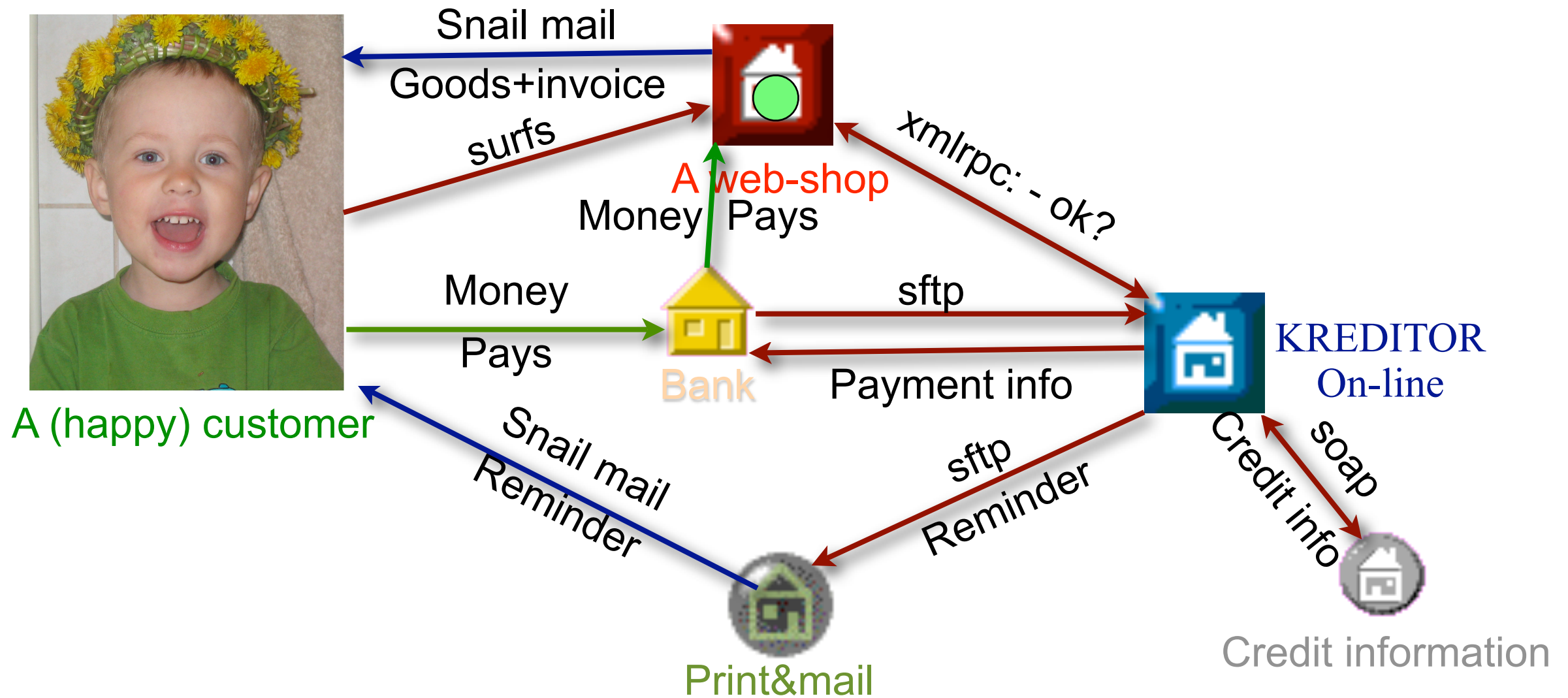
KREDITOR

Instalment Plans

- In **October 2007** we introduced instalment plans.
- That is, the ability to pay a (semi-) fixed amount in e.g. 24 month.
- Swedish law requires a written contract, so the previous market leader had a system requiring a signed contract, **taking minimum 3 days** to complete a purchase.
- We use **invoicing**, allowing the customer to choose an instalment plan without a contract; If he later sends in a contract he get the plan, otherwise he get an invoice. **The purchase is instantaneous.**
- In **February 2008** Kreditor became **the market leader** for instalment plans over Internet in Sweden.

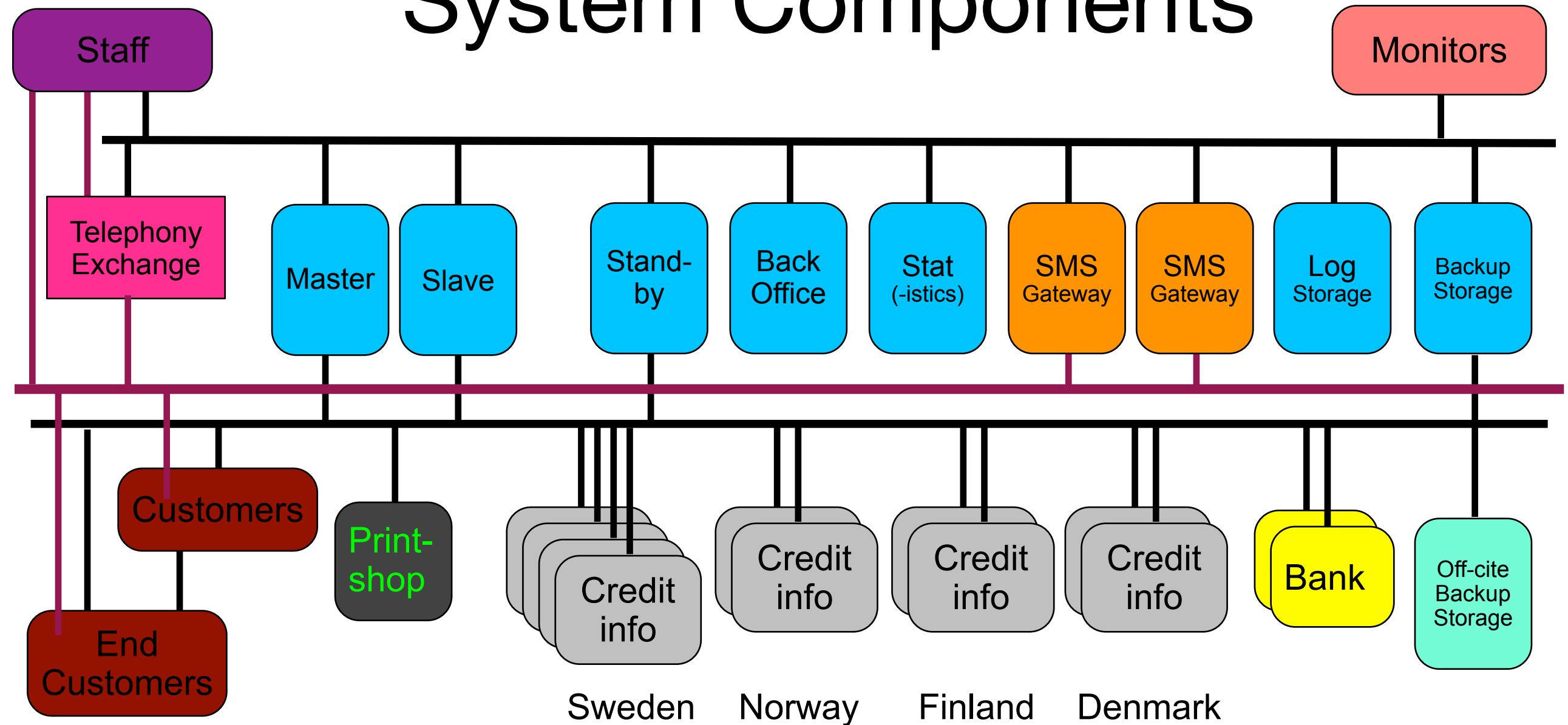
KREDITOR

How Does it Work?



KREDITOR

System Components



KREDITOR

Monitoring

- Demo

KREDITOR

Process

- We have have an informal **agile process**.
- We have very short **time to market**,
for simple changes the time from idea to finished integration
can be less than one hour.
- It is crucial for us to have an
automated comprehensive test suite.
 - With a framework that works.
 - Which is **used**.
 - **Always**.
- Enters **Yatsy** and **CruiseControl**.

KREDITOR

Yatsy & CC

- **Yet Another Test Server** – **Y**aws compatible.
(Yatsy is Swedish for Yatzee – testing is a bit like a dice game.)
- A home made test server, released: <http://code.google.com/p/yatsy/>
- **CruiseControl** is a framework for *continuous integration*.
- It is **open source**, it is written in **Java**, (but not by Kreditor).
- It automatically
 - checks out the latest version from a repository,
 - does a build,
 - runs all tests,
 - and it measures any metric you have automated.
 - As soon as **anyone** checks **anything** in.

KREDITOR

Our CC Set-up

- A set of projects
 - commit-[branch], runs on every commit
 - nightly-[branch], runs each night
- CC starts a script
 - svn update
 - make
 - run yatsy
- Reporting
 - Yatsy reports test results through an xml file
 - Reverse engineered from JUnit
 - CC sends email to responsible developer in case of error

KREDITOR

Integrating Cover with Yatsy

- There is an application called **Cover** in **Erlang/OTP** which can give you a line by line coverage analysis of a module.
- We have added a module `yatsy_cover` to **Yatsy** which will let you run cover on a test suite.
- If you run **yatsy** with the “cover flag” all beam files are “cover compiled” before the test suite is run.
- **Yatsy** produces a source code file annotated with execution count.

KREDITOR

The Erlang Advantage

- **E**asy to build fault-tolerant systems.
- **R**apid development.
- **L**ow-maintenance and easy upgrade.
- **A**bility to leverage multicore technology.
- **N**etwork programming is easy.
- **G**ood way to get great programmers.

KREDITOR

Easy to Build Fault-tolerant Systems

- **Erlang** was designed from the ground up with the purpose of making it easy to develop fault-tolerant systems.
- **Erlang** was developed by **Ericsson** with the telecom market in mind.
- **Erlang** supports processes, distributed systems, advanced exception handling, and signals.
- **Erlang** comes with **OTP**-libraries (**Open Telecom Platform**), e.g. *supervisors* and *generic servers*.

KREDITOR

Rapid Development

- **Erlang** has a number of features to support rapid prototyping and fast development:
 - Automatic memory management.
 - Symbolic constants (atoms).
 - An interactive shell.
 - Dynamic typing.
 - Simple but powerful data types.
 - Higher order functions and list comprehensions.
 - Built in (distributed) database.

KREDITOR

Low-maintenance and Easy Upgrade

- **Erlang** has a number of features that makes it easy to maintain and upgrade:
 - Hot code loading.
 - Distribution.
 - Interactive shell.
 - Simple module system.
 - No shared state.
 - Virtual machine.

KREDITOR

Ability to Leverage Multi Core

- The concept of processes is an integral part of **Erlang**.
- The Erlang Virtual machine (**BEAM**) has support for *symmetric multiprocessing*.
- No shared memory -- easier to program.

KREDITOR

Network Programming is Easy

- Distributed **Erlang** solves many network programming needs.
- Setting up a simple socket protocol is a breeze.
- The binary- (and now bit-) syntax makes parsing binary protocols easy.
- There are simple but powerful libraries for HTTP, XML, XML-RPC and SOAP.

KREDITOR

Good Way to Get Great Programmers

- **Nice paradox:**
The lack of **Erlang** programmers makes it easier for us to find great programmers.
- There are many great C and Java programmers, I'm sure, but they are hidden by hordes of mediocre programmers.
- Programmers who know a functional programming language are often **passionate** about programming.
- Passionate programmers makes **Great Programmers™**.

KREDITOR

Conclusion

A good business plan
and
a great programming language
is
a powerful combination.

KREDITOR