

GOLEM IV

Invisible Banking

by

Bzowska, Ewa

Deptak, Victor

Pepliński, Szymon P.

Żak, Piotr



**“The question is not only
what bank can do.
It's what bank should do.”**

The big idea of

“GOLEM IV – Invisible Banking”

Invisible bank is not like other banks.

We know banking isn't something people want to spend all their time thinking about.

Not in a world that has billion other interesting things to do. We believe everyone deserves a bank that enables them to improve their future, with minimum friction and maximum ease. We focused on the growing need to automate time consuming payments, in a way that's simple, safe and convenient.

GOLEM IV – Invisible Banking

Kate – Blind

- problems with handling payments
- needs help from others

Tom, Ann & Amy – Family

- they spend a lot of time paying bills instead of spending time together

John – Car Mechanic, Business Owner

- sometimes he forgot to pay the invoice
- spends too much time accounting



The Distribution of Users' Computer Skills: Worse Than You Think

[nngroup.com/articles/computer-skill-levels](https://www.nngroup.com/articles/computer-skill-levels)

26% of Adult Population = Can't Use Computers

62% of Adult Population = Can't Use Your Designs

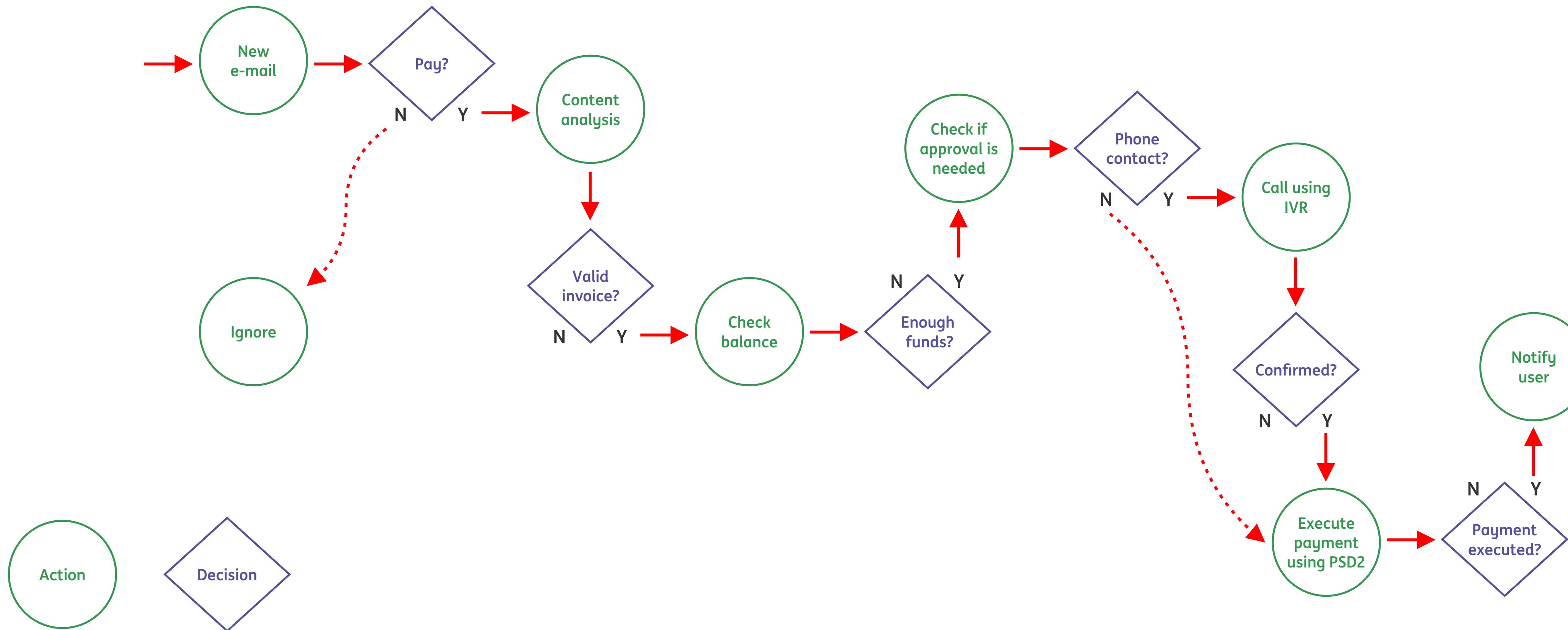
GOLEM IV – Invisible Banking

How does it work?

- 1 GOLEM IV creates an order based on an email message (e.g., attached invoice)
- 2 AI/ML model verifies the authenticity of the document
- 3 GOLEM IV decides if there's a need for additional confirmation; if so, it calls using the IVR system
- 4 GOLEM IV carries out payments using PSD2 (Open Banking) interfaces

In most cases, the entire process is automatic (no interface required)

Solution Flow



GOLEM IV – Invisible Banking

AI/ML Models

github.com/PiotrZak/lem-golem

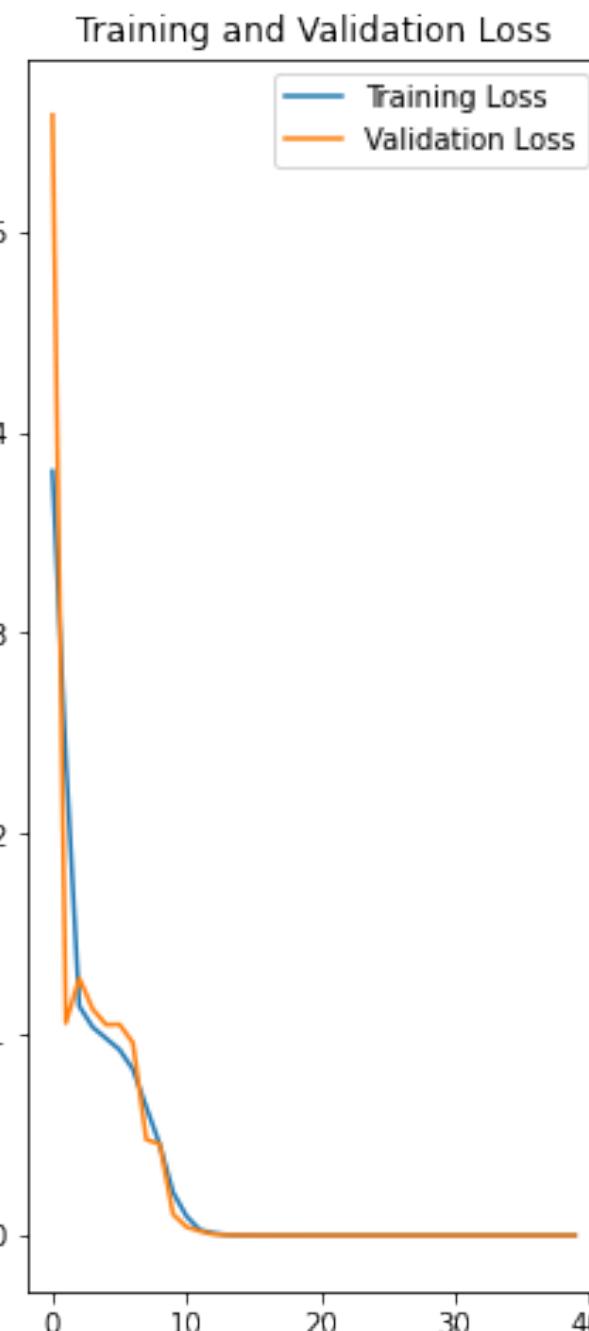
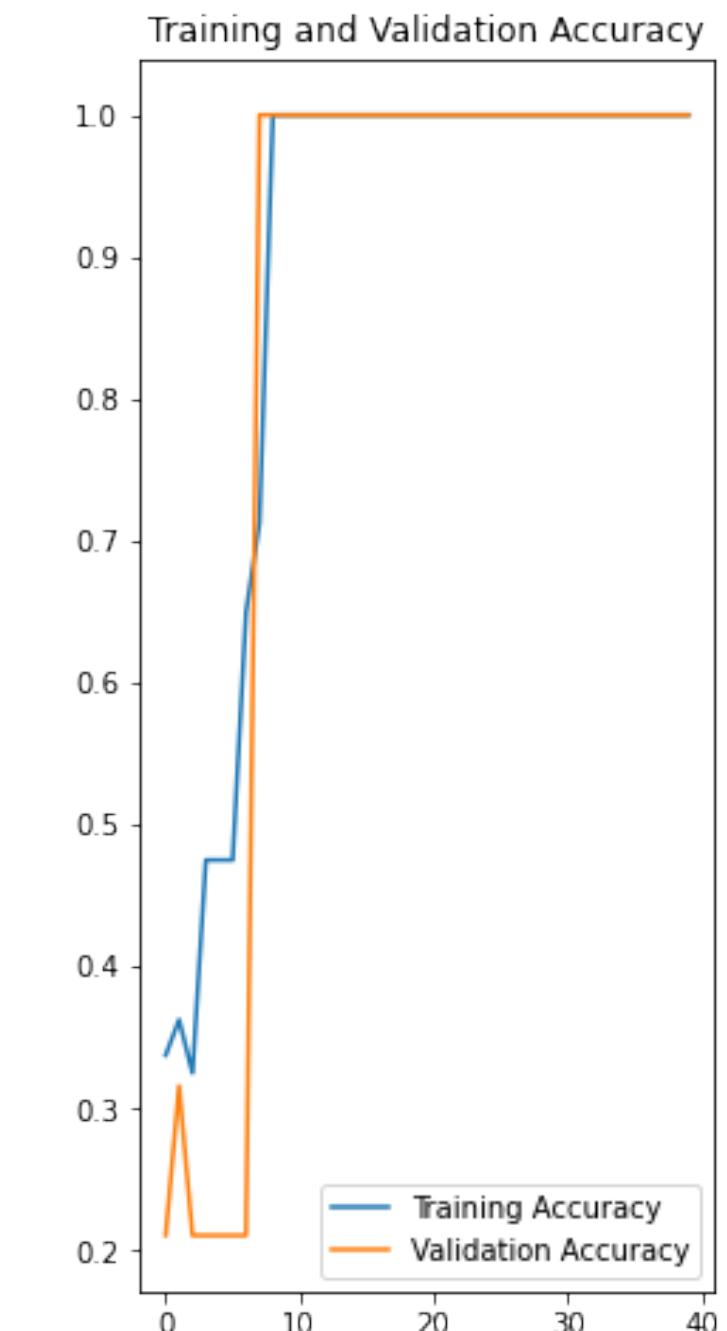
This invoice most likely belongs to
PLUS with a 100% confidence.



This invoice most likely belongs to
PKO LEASING with a 100% confidence.



Training results



GOLEM IV – Invisible Banking

Integrations

github.com/PiotrZak/lem-golem

Open Banking (PSD2) with Revolut

```
6 import csv
7 from datetime import datetime
8 import io
9
10 from revolut import Amount, Transaction
11
12 _CSV_COLUMNS = ["date", "hour", "from_amount", "from_currency",
13                 "to_amount", "to_currency"]
14
15
16 def csv_to_dict(csv_str, separator=","):
17     """ From a csv string, returns a list of dictionnaries
18     """
19     [{}]
20     [{}]
21     [{}]
22     [{}]
23     [{}]
24     reader = csv.DictReader(io.StringIO(csv_str), delimiter=separator)
25
26     # By default, DictReader returns OrderedDict => convert to dict:
27     return list(map(dict, reader))
28
29
30 def append_dict_to_csv(filename, dict_obj, separator=",",
31                       col_names=_CSV_COLUMNS):
32     """ Append a dict object, to a csv file """
33     with open(filename, 'a', newline='\n') as csvfile:
34         writer = csv.DictWriter(csvfile,
35                               delimiter=separator,
36                               fieldnames=col_names,
37                               lineterminator='\n') # To avoid '^M'
38         writer.writerow(dict_obj)
39
40
41 def convert_Transaction_to_dict(transaction_obj):
42     return {
43         "date": transaction_obj.date.strftime("%d/%m/%Y"),
44         "hour": transaction_obj.date.strftime("%H:%M:%S"),
45         "from_amount": transaction_obj.from_amount.real_amount,
46         "from_currency": transaction_obj.from_amount.currency,
47         "to_amount": transaction_obj.to_amount.real_amount,
48         "to_currency": transaction_obj.to_amount.currency,
49     }
```

Mailbox automation

- Support for SMTP AUTH (RFC4954) and PIPELINING (RFC2920)
- Web interface to view messages (plain text, HTML or source)
 - Supports RFC2047 encoded headers
- Real-time updates using EventSource
- Release messages to real SMTP servers
- Chaos Monkey for failure testing
 - See [Introduction to Jim](#) for more information
- HTTP API to list, retrieve and delete messages
 - See [APIv1](#) and [APIv2](#) documentation for more information
- HTTP basic authentication for MailHog UI and API
- Multipart MIME support
- Download individual MIME parts
- In-memory message storage
- MongoDB and file based storage for message persistence
- Lightweight and portable
- No installation required

sendmail

mhsendmail is a sendmail replacement for MailHog.

It redirects mail to MailHog using SMTP.

You can also use `MailHog sendmail ...` instead of the separate mhsendmail binary.

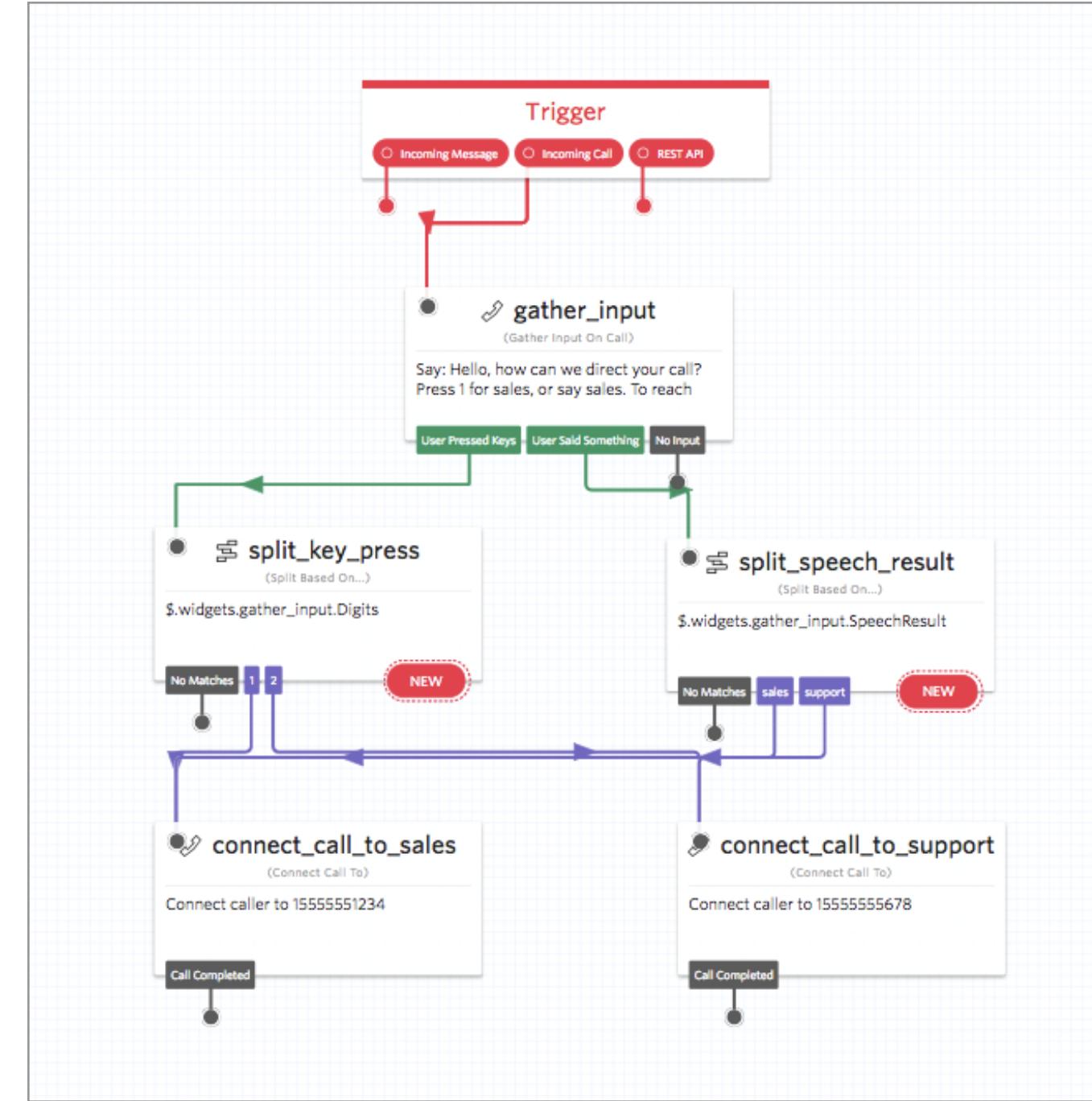
Alternatively, you can use your native `sendmail` command by providing `-S`, for example:

```
/usr/sbin/sendmail -S mail:1025
```

For example, in PHP you could add either of these lines to `php.ini`:

```
sendmail_path = /usr/local/bin/mhsendmail
sendmail_path = /usr/sbin/sendmail -S mail:1025
```

Build an IVR with Twilio



Benefits

- AI/ML verifies documents and metadata, minimizing the risk of attack (e.g. phishing) or making a mistake
- No visual interface (UI) – effective solutions for the elderly, people with lower digital skills, or those using Assistive Technologies
- The user doesn't have to remember about payments
- Time saving

The source of income

- A free solution for retail customers (several payments a month)
- B paid solution for business clients (professionals / SME / MC) – subscription-based or bundled model