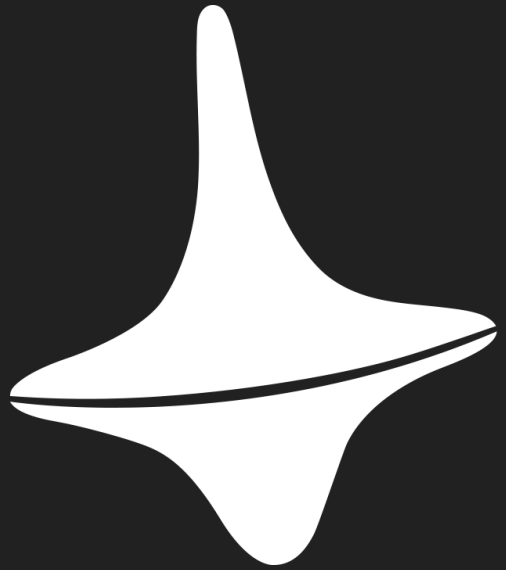
Thank you,

Ruby World 2025

Bernard Banta

@Bantab



I work at

Finplus Group

CTO and Co-Founder

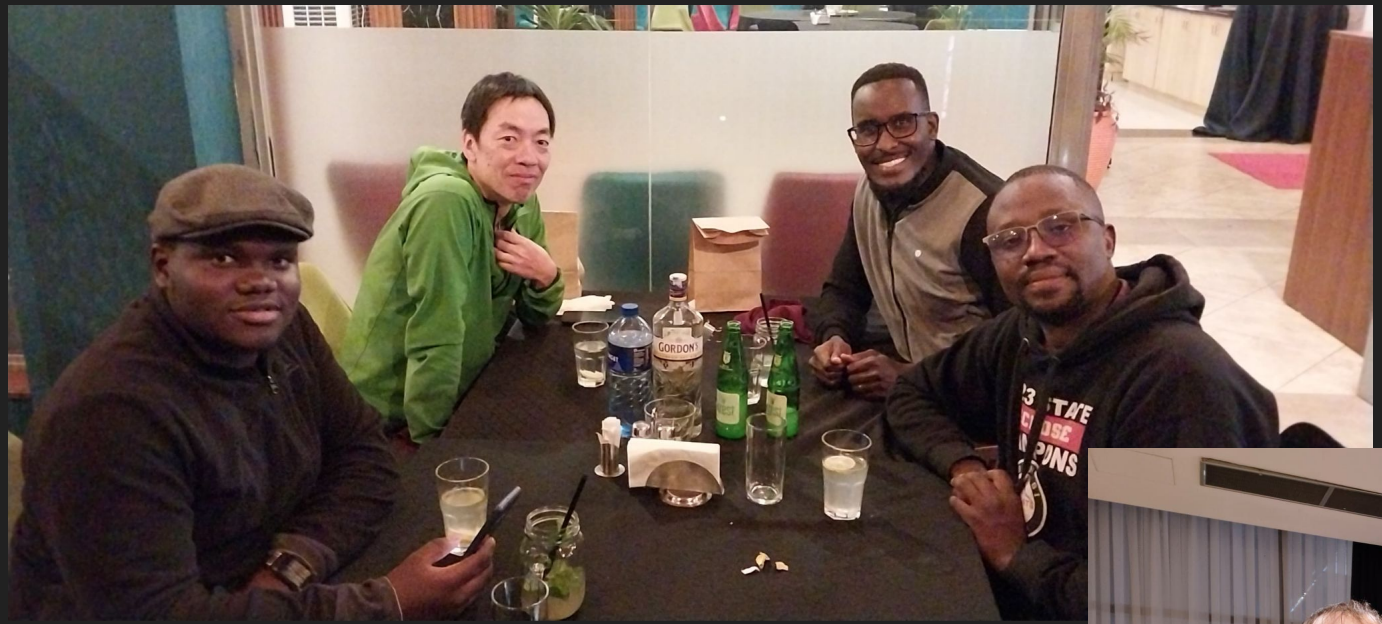
ARC

Chairman and Co-founder

rubycommunity.africa

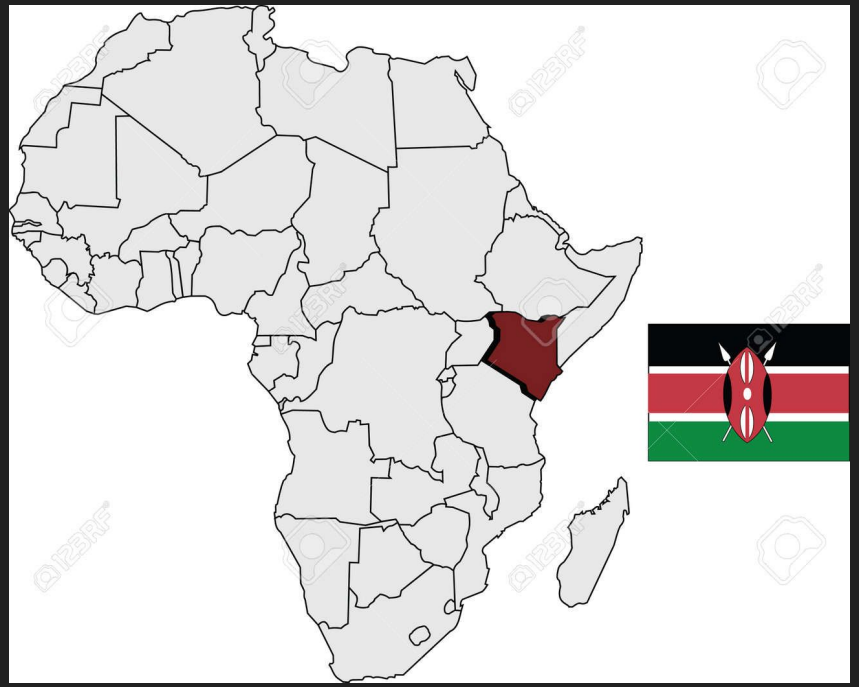

RubyConf


Share your Feedback
Ruby Conf
Africa ®


GORDON'S

I’m from

Nairobi, Kenya




Olga Levari Ercolano

rubyconf.africa

RubyConf Thai23: Breaking Barriers

Empowering the Unbanked with Innovative Tech

Globally, some 1.4 billion adults remain unbanked. These people are hardest to reach – and more commonly women, poorer, less educated, and living in rural areas.

Understanding the Challenges

- Limited access to banking infrastructure
- Low income
- Digital illiteracy
- High transaction costs
- Geographical barriers
- Limited connectivity
- Informal work

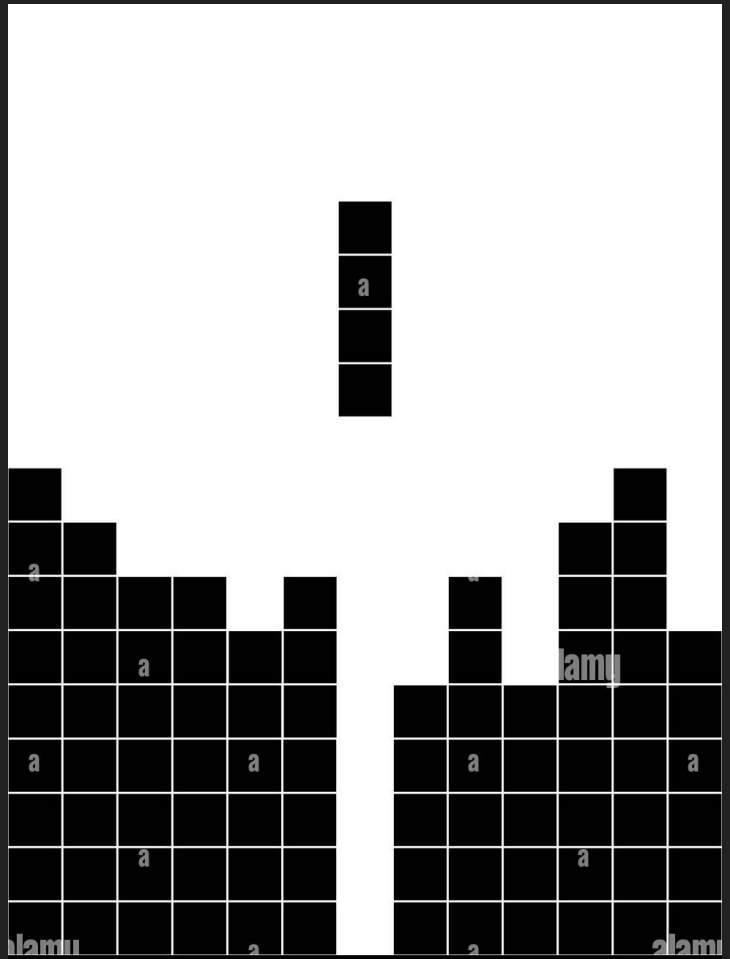
What phones do they use;

- Most of them use feature phones
- Minority of them use low-end smartphones with minimal internet access


08:59
2021/11/19 Friday
menu
contacts

What can you do with a feature phone?


SNAKE


alamy
Image ID: 2CDR7YW
www.alamy.com

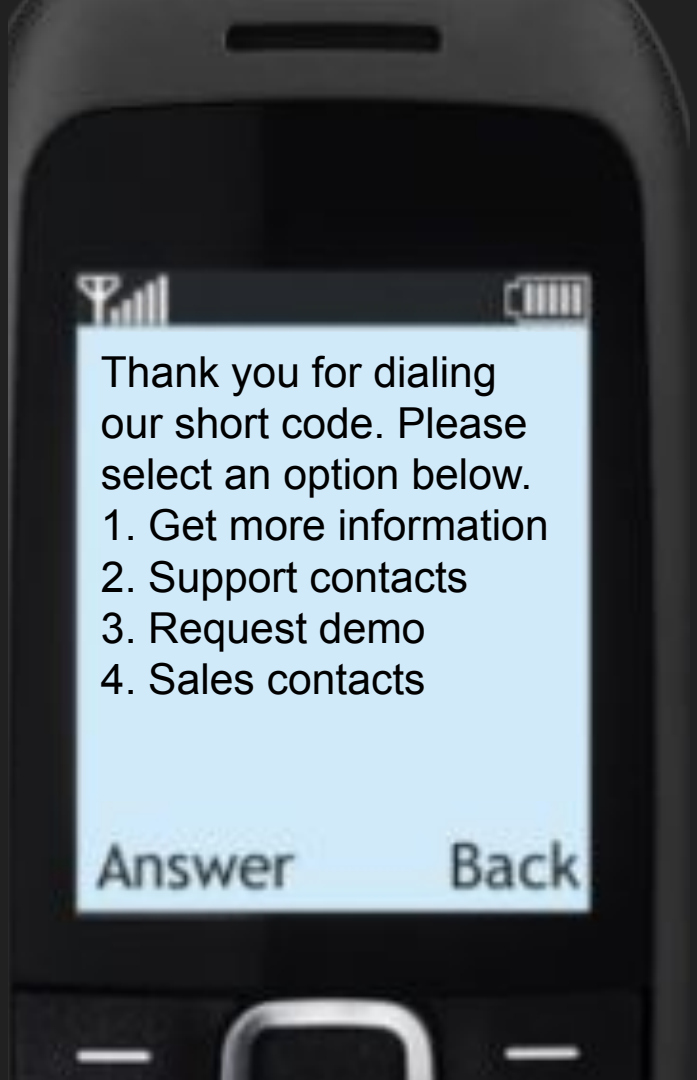

USSD

Feature phone

Telecommunication company

Rails backend server

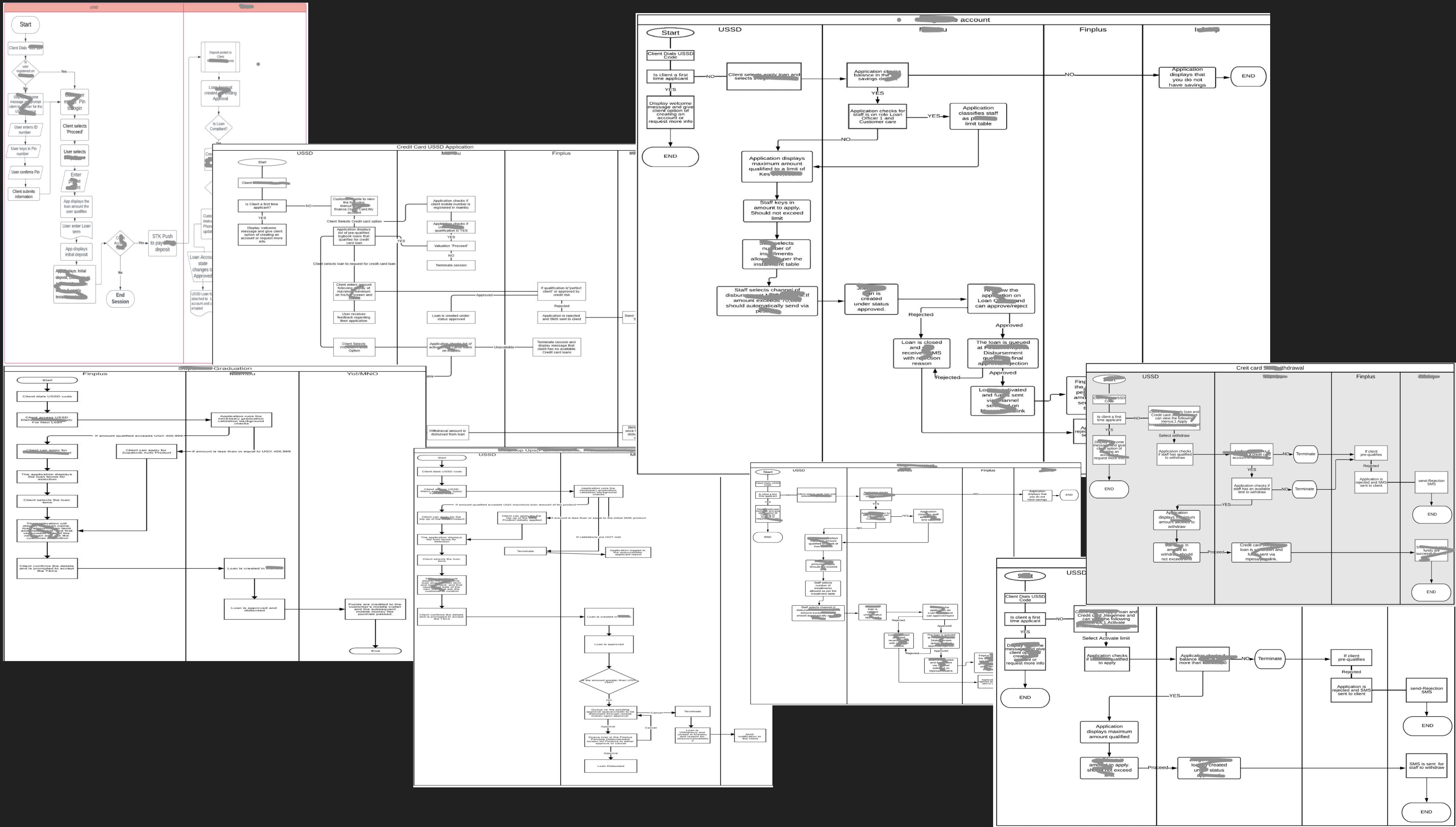
How do you develop a USSD app?



For a simple app you can use conditional statements

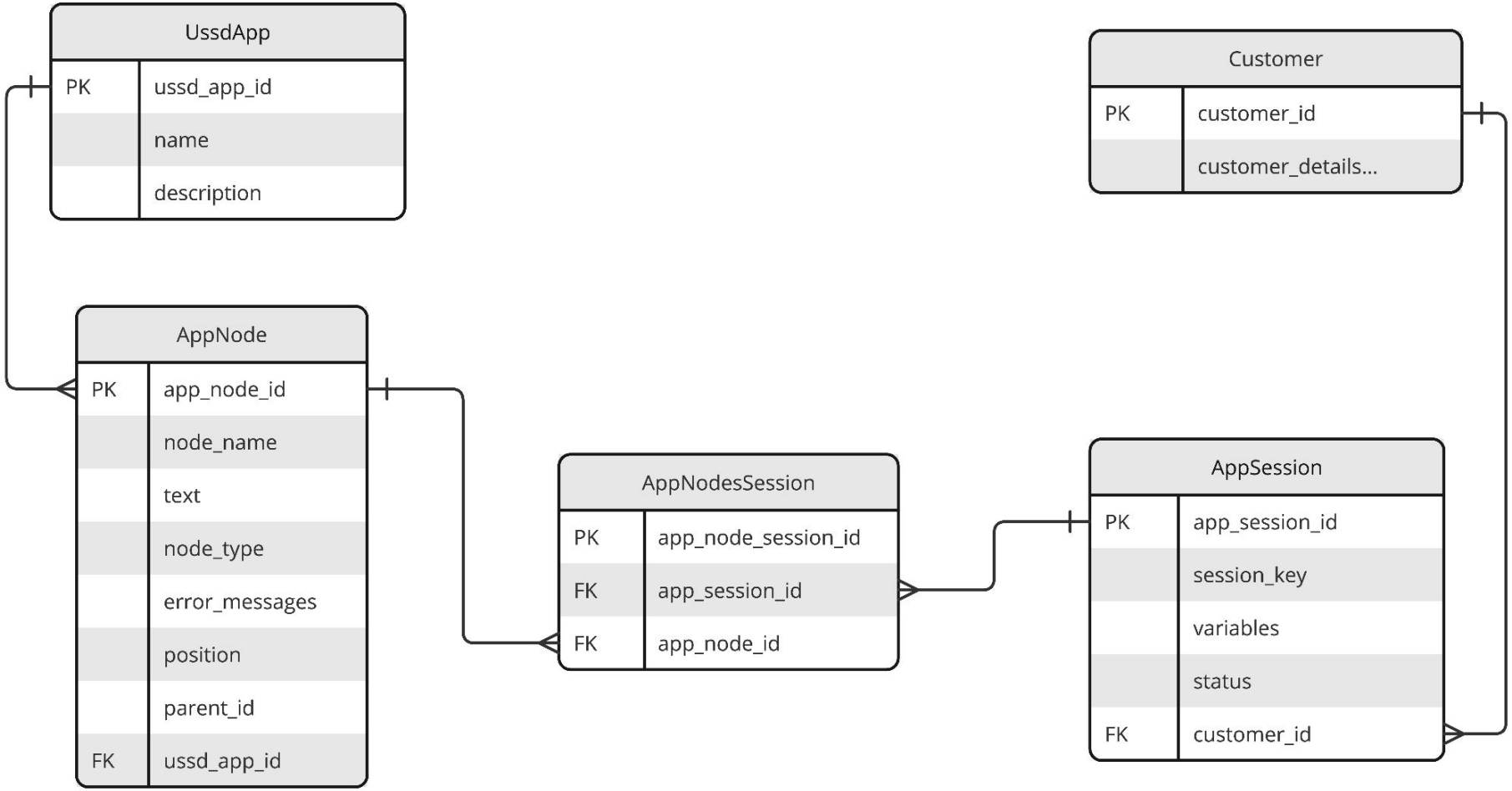
```ruby
case user_input
when '1'
  # Send an SMS about more information
when '2'
  # Send an SMS about support contacts
when '3'
  # Log demo request
when '4'
  # Send an SMS about sales contacts
else
  # Display invalid input
end
```

How about a complex app?



USSD
Start
Client Dials USSD Code
Is client a first time applicant
YES
NO
Display welcome message and give client option of creating an account or request more info
END
Application displays that you do not have savings
Application checks for staff is on role Loan Officer 1 and Customer care
Application classifies staff
limit table
Application displays maximum amount qualified to a limit of
Staff keys in amount to apply. Should not exceed limit
Staff selects channel of
Rejected
Approved
Finplus
Credit Card USSD Application
Mambu
Is Client a first time applicant?
Application checks if client mobile number is registered in mambu
Valuation "Proceed"
Terminate session
Client Selects Credit card option
Client selects loan to request for credit card loan
Loan is created under status approved
Application is rejected and SMS sent to client
Terminate session and display message that client has no available Credit card loans
Unavailable
Graduation
Yo!/MNO
Client dials USSD code
Application runs the necessary graduation validation background checks
If amount qualified exceeds UGX 400,999
If amount is less than or equal to UGX 400,999
The application displays the loan terms for selection
Client selects the loan term
Client confirms the details and is prompted to accept the T&Cs
Loan is approved and disbursed
Funds are credited to the customer's mobile wallet and the subsequent mobile money fee journals passed.
End
Loan is approved
Loan Disbursed
Terminate
Creit card withdrawal
Select withdraw
Terminate
If client pre-qualifies
Application is rejected and SMS sent to client
send-Rejection SMS
Select Activate limit
Application displays maximum amount qualified
Proceed
SMS is sent for staff to withdraw
STK Push
End Session
Is Loan Compliant?

We came up with a Component-Based solution approach

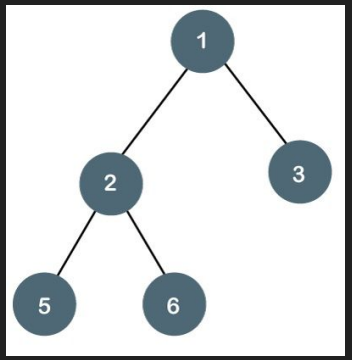

UssdApp
PK
ussd_app_id
name
description
Customer
PK
customer_id
customer_details...
AppNode
PK
app_node_id
node_name
text
node_type
error_messages
position
parent_id
FK
ussd_app_id
AppNodesSession
PK
app_node_session_id
FK
app_session_id
FK
app_node_id
AppSession
PK
app_session_id
session_key
variables
status
FK
customer_id

We create a USSD tree file in JSON format compatible with the AppNode model structure we saw previously.

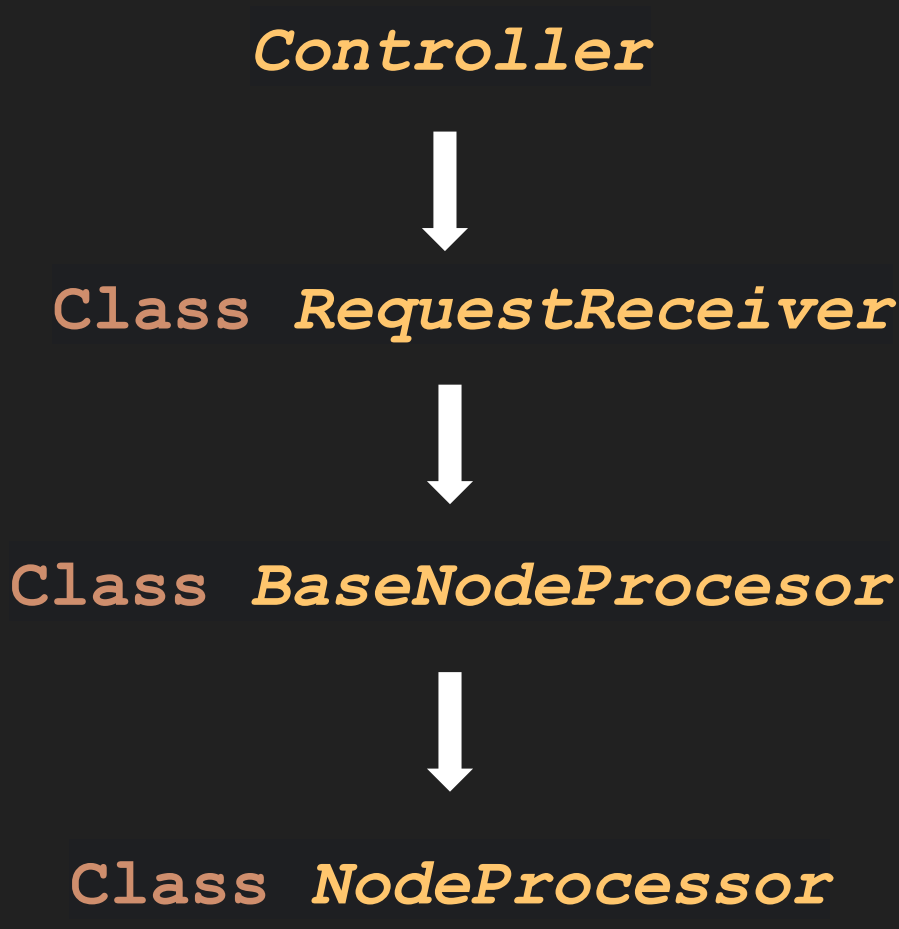
Lending USSD App

Generate tree

- Root Node
  - new_client {4403, 4404}
    - register {4404, 4405}
      - mother_maiden_name {4405, 4406}
      - primary_school_name {4405, 4407}
      - official_email_address {4405, 4408}
      - enter_new_pin {4405, 4409}
        - confirm_pin {4409, 4410}
    - request_more_info {4404, 4411}
    - loans_leads {4404, 4412}
      - loans_lead_product {4412, 4413}
        - asset_financing_selection {4413, 4414}
        - car_import_financing_selection {4413, 4415}
        - logbook_loans {4413, 4416}
        - civil_servants_loans {4413, 4417}
        - sme_two_loans {4413, 4418}
        - banker_loans {4413, 4419}
        - landlord {4413, 4420}
        - stock_loan {4413, 4421}
        - [redacted]_loan {4413, 4422}
        - import_finance {4413, 4423}
        - land_financing {4413, 4424}
        - [redacted]loan {4413, 4425}
        - client_location {4413, 4426}
        - consent {4413, 4427}
        - [redacted]_phone_loan {4413, 4428}
          - easy_phone_amount {4428, 4429}
    - redeem_airtime {4404, 4430}
    - car_bid {4404, 4431}
      - car_bid_email_address {4431, 4432}
        - car_bid_registration {4432, 4433}



We generate a tree on the DB based on the JSON file.


Controller
Class RequestReceiver
Class BaseNodeProcesor
Class NodeProcessor

```ruby
class BaseProcessor
  attr_reader :request, :app, :session, :current_node, :client, :user_input, :phone_number

  class << self
    def call(request)
      new(request).process_request
    end
  end

  def initialize(request) → void
    @request = request
    @session = request.session
    @current_node = session.current_node
    @phone_number = request.phone_number
    @client = request.client
    @user_input = request.input
    @app = request.app
  end

  def process_request
    raise 'Not implemented'
  end

  private

  def set_current_node(node_name)
    session.set_current_node(node_name)
  end

  def current_node_prompt
    session.current_node.prompt
  end

  ## Other methods
end
```

```ruby
class NodeProcessor1 < BaseProcessor
 def process_request
   # Implement your logic here
 end
end
```

What more can you do with USSD?

- Web browser and
- Mobile USSD apps

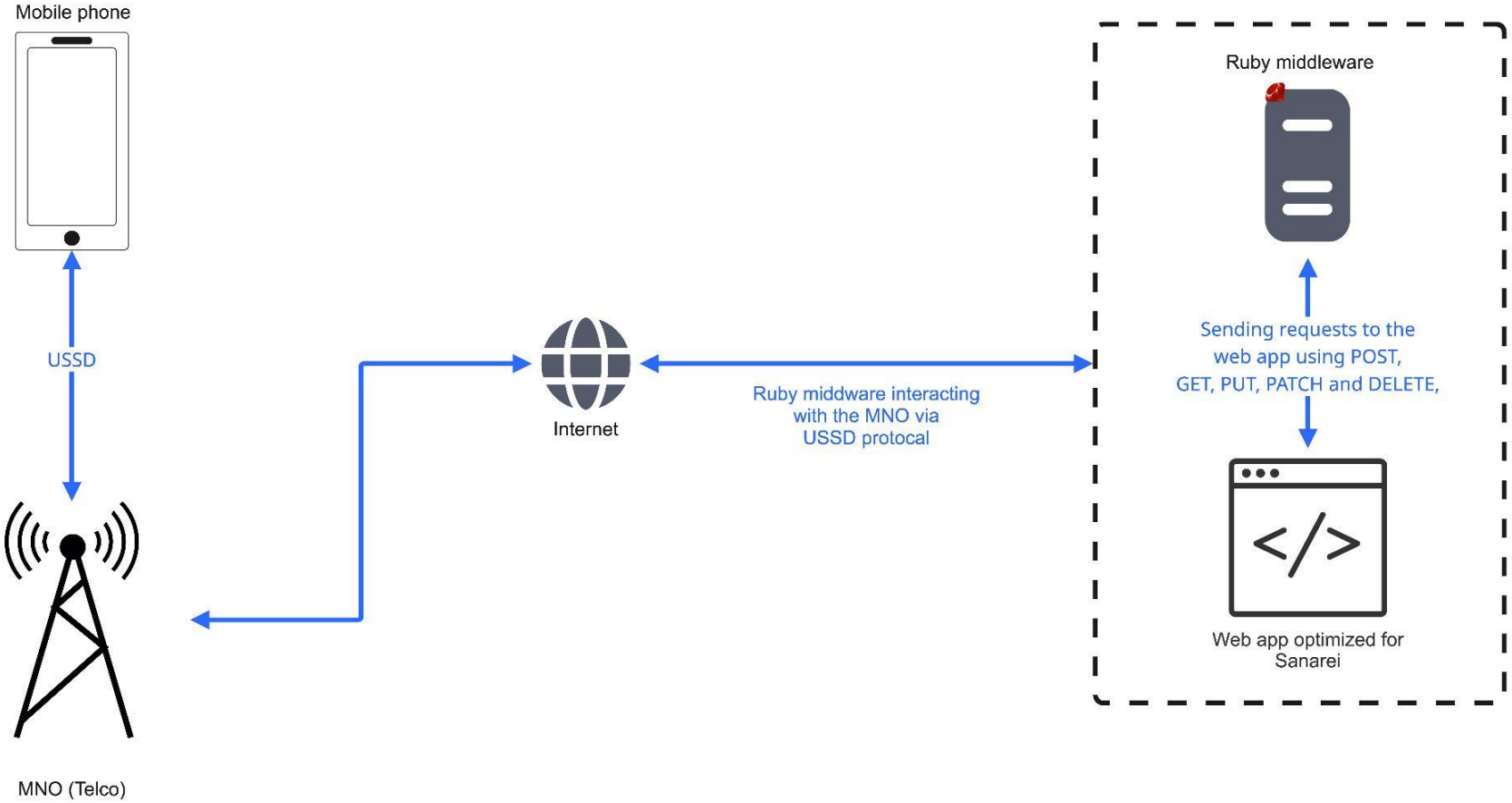
Offline Web Browsing

Why is it important?

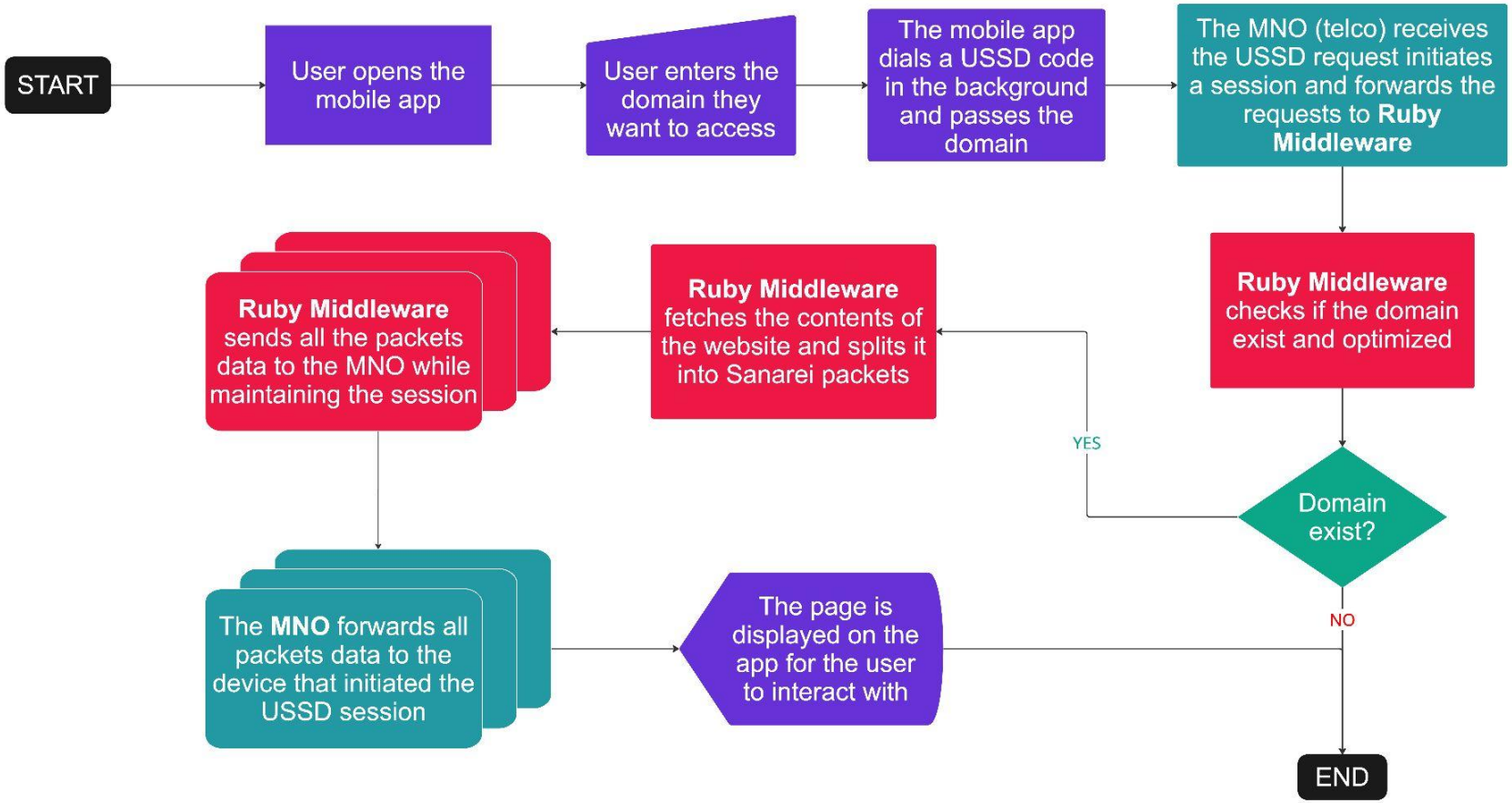

Sanarei

How does it work?

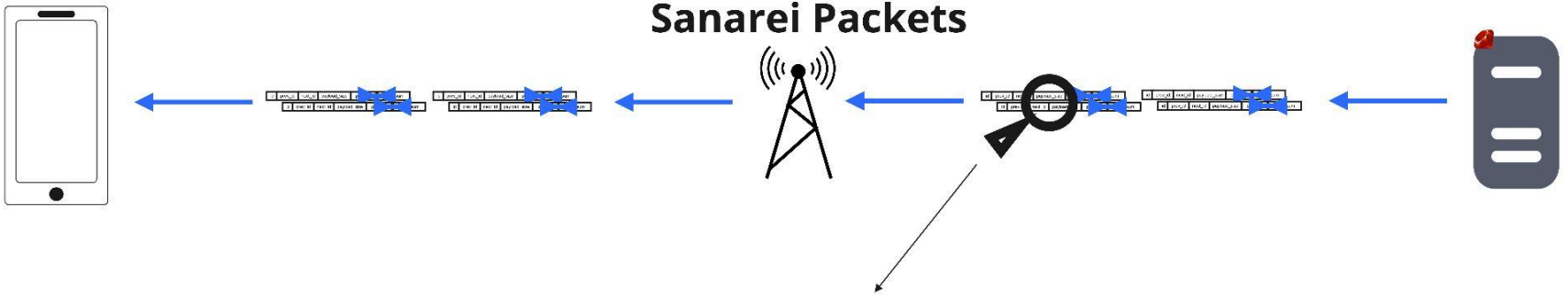
Sanarei Architecture



Sanarei Web Browsing Flowchart



Sanarei Packets



id	prev_id	next_id	payload_size	payload	checksum

Packet details

Field	Type	Bytes	Description
id	unsigned short (S)	2	Packet number (1, 2, 3…)
prev_id	unsigned short (S)	2	Previous packet ID (0 if none)
next_id	unsigned short (S)	2	Next packet ID (0 if none)
payload_size	unsigned int (L)	4	Number of bytes in payload
payload	bytes	140	Compressed data chunk
checksum	unsigned short (S)	2	Optional CRC16 checksum

Short demo

github.com/orgs/sanarei/repositories

- **Website:** sanarei.com
- **Mobile app:** sanarei-kotlin-app
- **Ruby middleware:** sanarei-middleware
- **Sample webapp:** sanarei-sample-app

Limitations

Next steps

- USSD VPN proxy
- Are there other ways (LoRa)
- Develop the apps for feature phones

What other problems can USSD solve?


Thank you