# Hi!

# The opposite of every great idea is another great idea.
# – Niels Bohr

# SMTP email is push.

SMTP is the opposite sh.

The opposite of every great idea is another great idea.
– Niels Bohr

# Maybe email could be pull.

**Sender ISP** → **Receiver ISP**

☺

☹
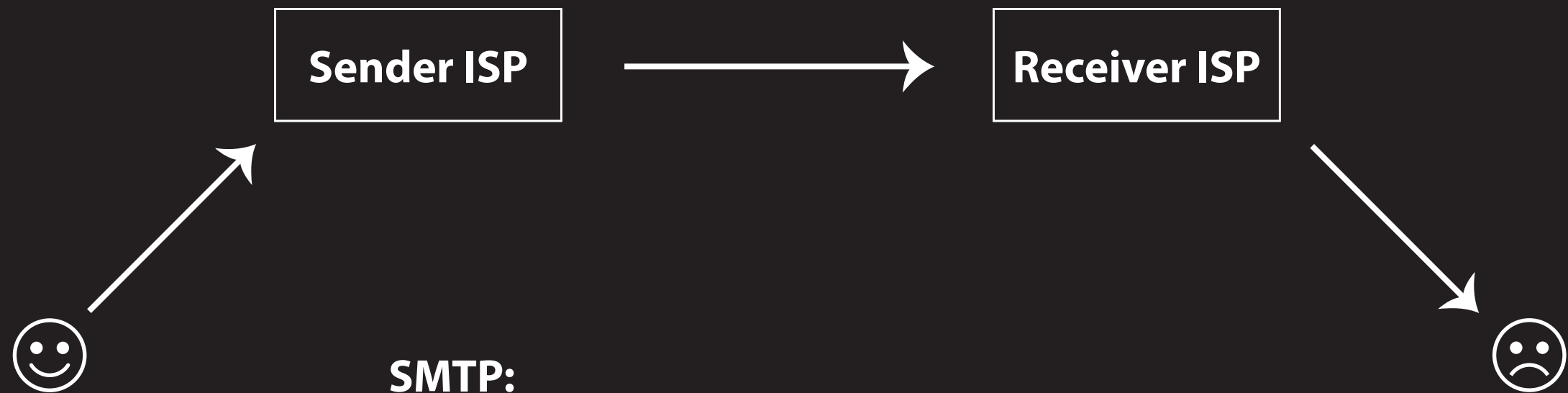
**SMTP:**
    **You POP against the receiver ISP**

**Stub:**
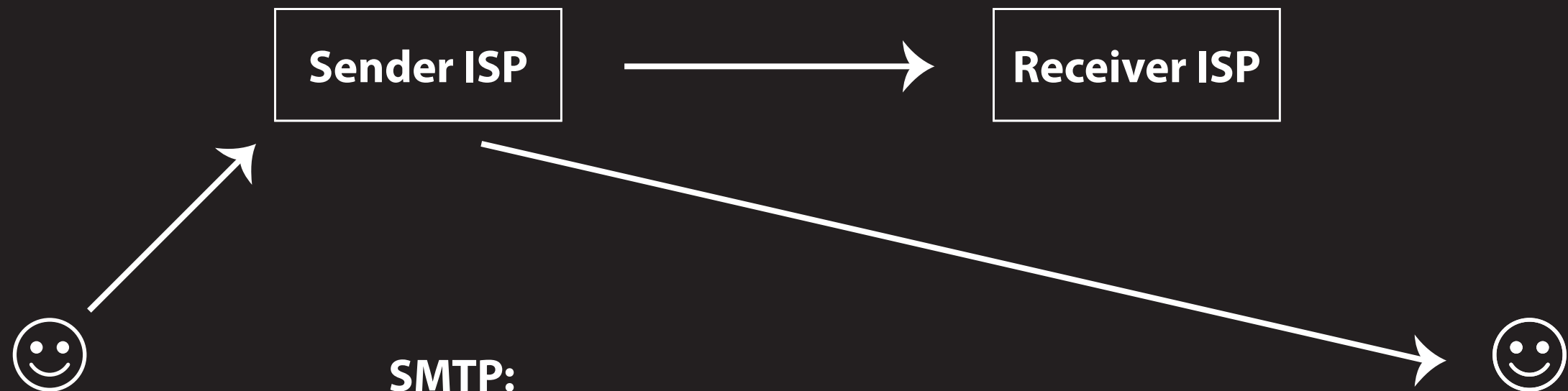    **You POP against the sender ISP**

# The opposite of every great idea is another great idea. — Niels Bohr

Sender ISP → Receiver ISP

SMTP:
You POP against the receiver ISP

You POP against the sender ISP

**Sender ISP** → **Receiver ISP**

**SMTP:**
   **You POP against the receiver ISP**

**Stub:**
   **You POP against the sender ISP**

# how does this solve spam?

# In a pull world, you do not hear:

# "I keep going to this website against my will!"

# "OMG this blog won't stop sending me new posts"

# But in a push world,

# "Dear Mailing List,

# Please stop sending me stuff!"

# "Dear Mailing List,

# Please stop sending me stuff!"

# … WRONG!

# "Dear Mailing List,

# Please stop sending me stuff!"

# ... WRONG!

# Power is in the wrong place.

Sender ISP → Receiver ISP

☺

**Senders maintain a list of receivers.** ☹

*addressbook*

— ☺ —
— ☺ —
— ☺ —

# The opposite of every great idea is another great idea.
## – Niels Bohr

**Sender ISP** → **Receiver ISP**

# Receivers maintain a list of senders!

*addressbook*

**Sender ISP** → **Receiver ISP**

😊

**Receivers maintain a list of senders!**

**Just like RSS – but one-to-one!**

😊

*addressbook*
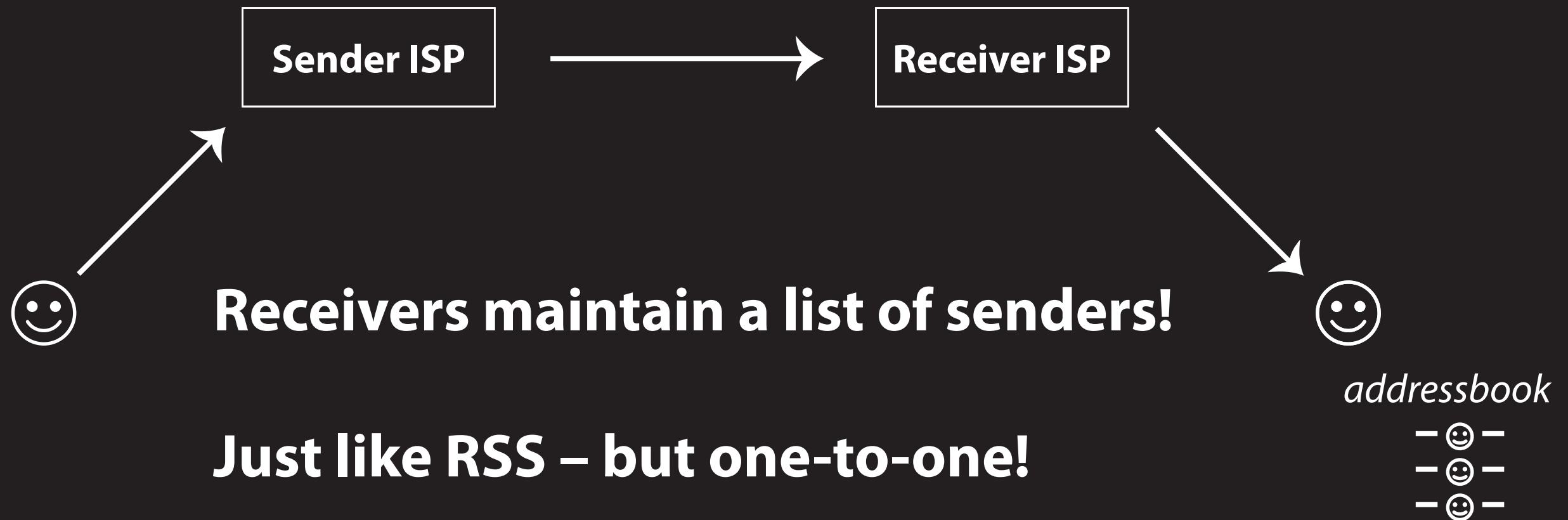
😊
😊
😊
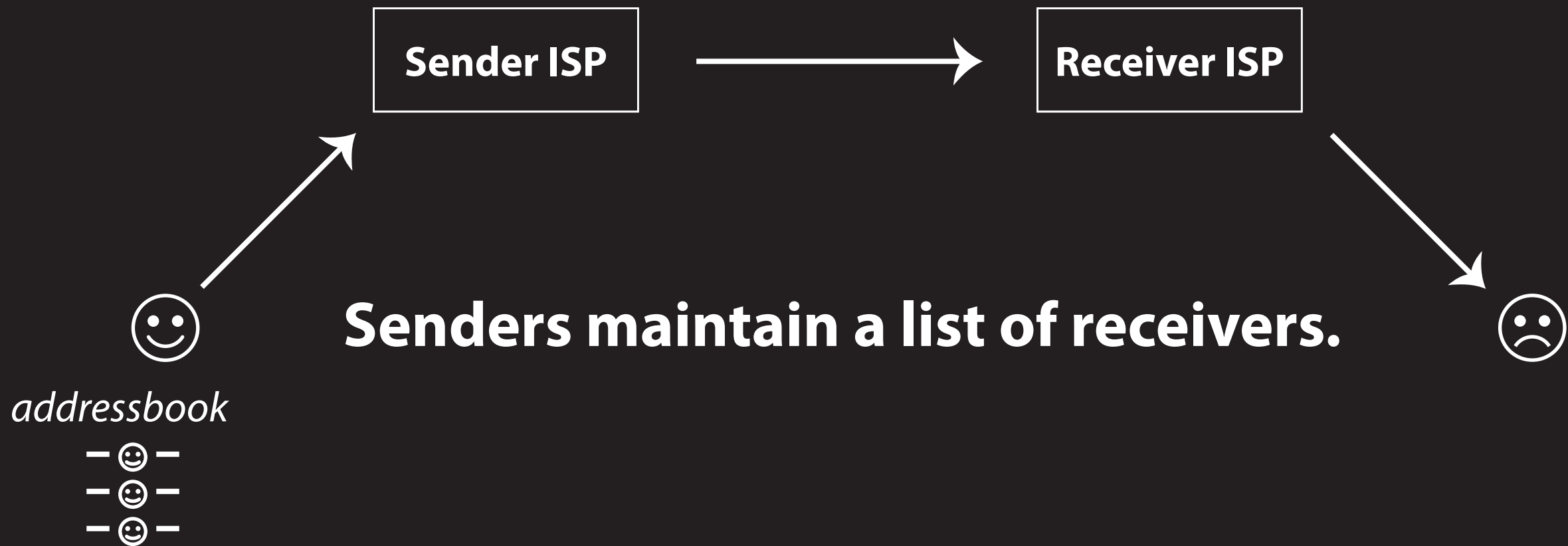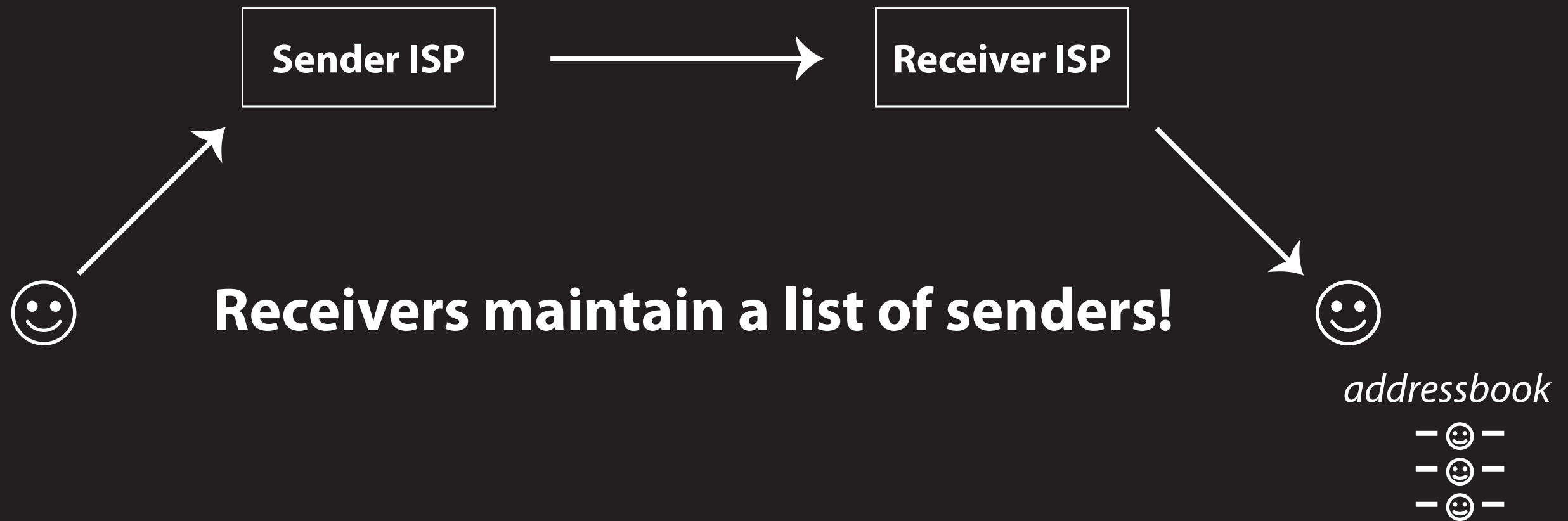
# DJB had the idea first.

## IM2000

# It'll never work!

When a distinguished but elderly scientist states that something is possible, he is almost certainly right.

When he states that something is impossible, he is very probably wrong.

– Arthur C Clarke

# April 8 2006.
# Meng hosted a hackathon.

- **Julian Haight (SpamCop)**
- **Nathan Cheng (HTMP)**
- **Richard Soderberg (Coral)**
- **Meng Weng Wong (SPF, Pobox)**

# The Goal:
Meng and Julian can email each other without worrying about spam filtering.

# Not The Goal:
# Get everybody in the world to switch.

# Not The Goal:
**Convince grouchy gray-beards to stop using email.**

# Email will go on.

# This is just another thing.

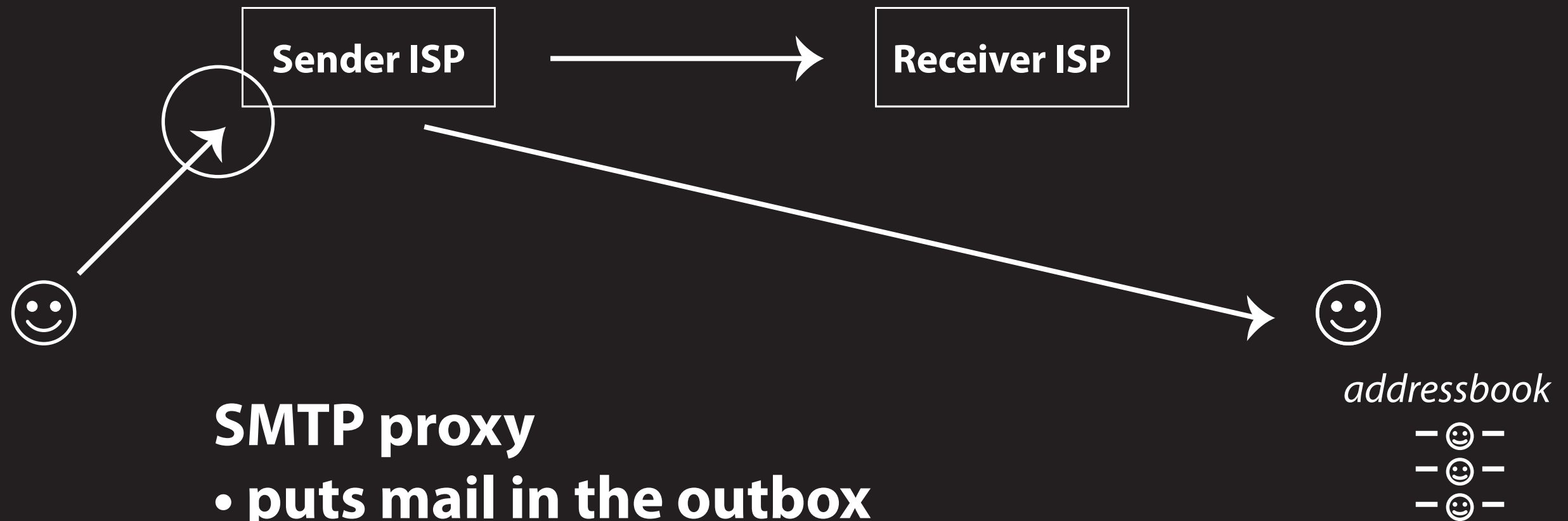# Email will go on.

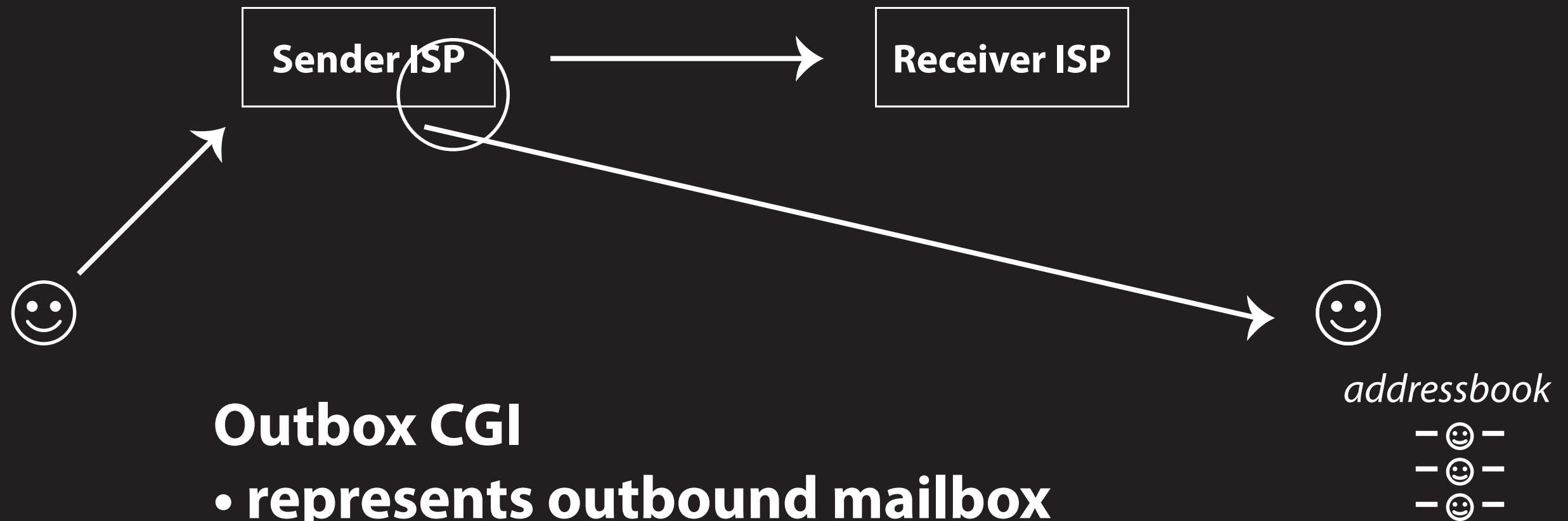# If spam can't kill it, what makes you think this will?

# July 18 2006.
# Meng writes to Julian:

*Come here, Watson, I need to tell you that one small step for (a) man is one giant leap for mankind.*

**SMTP proxy**
- **puts mail in the outbox**
- **knows how to fail over to SMTP**

**Sender ISP** → **Receiver ISP**

*addressbook*

**Outbox CGI**
- **represents outbound mailbox**
- **handles queries from clients**

**Sender ISP** → **Receiver ISP**
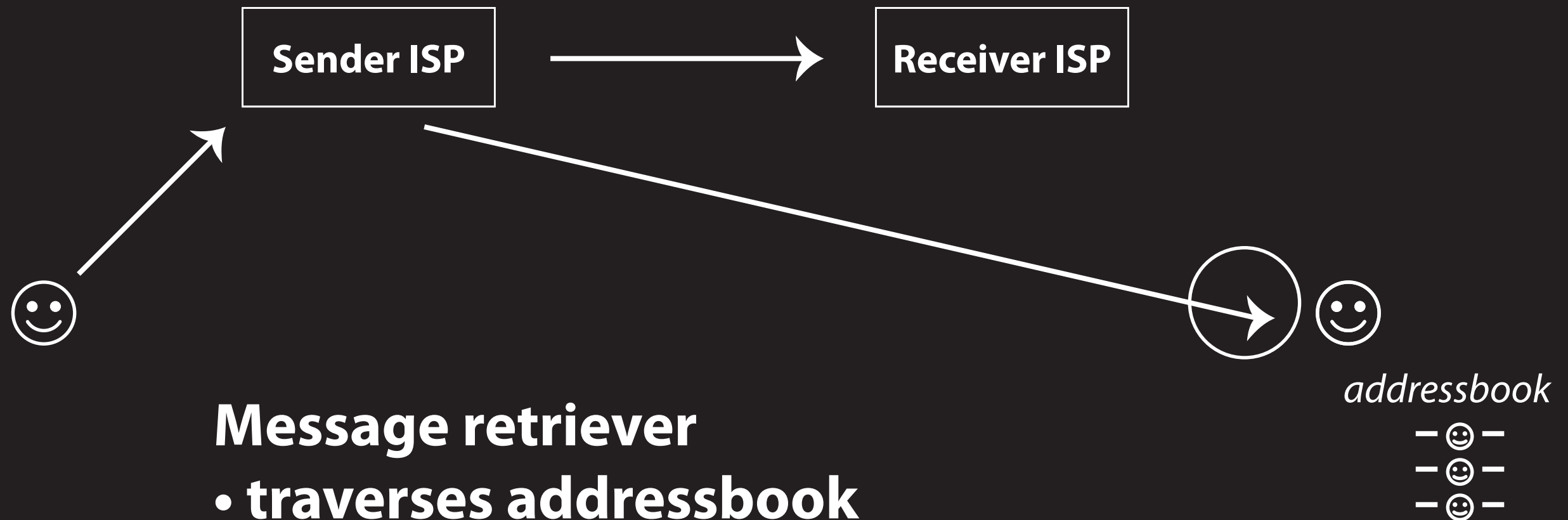
*addressbook*

## Message retriever
- **traverses addressbook**
- **sucks down new mail**
- **dumps it in your inbox**

# *But why "Stubmail"?*
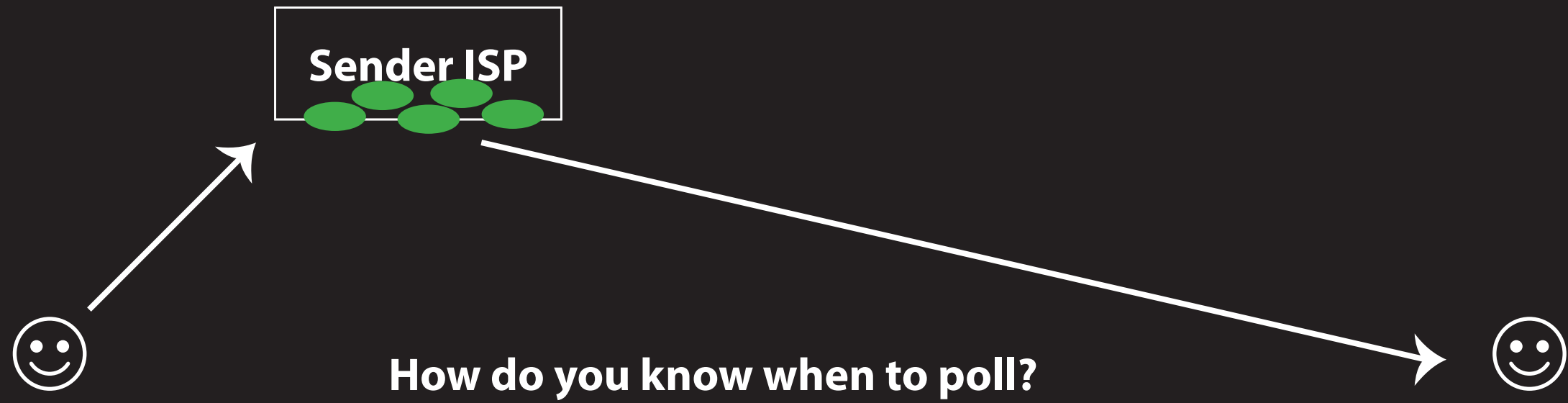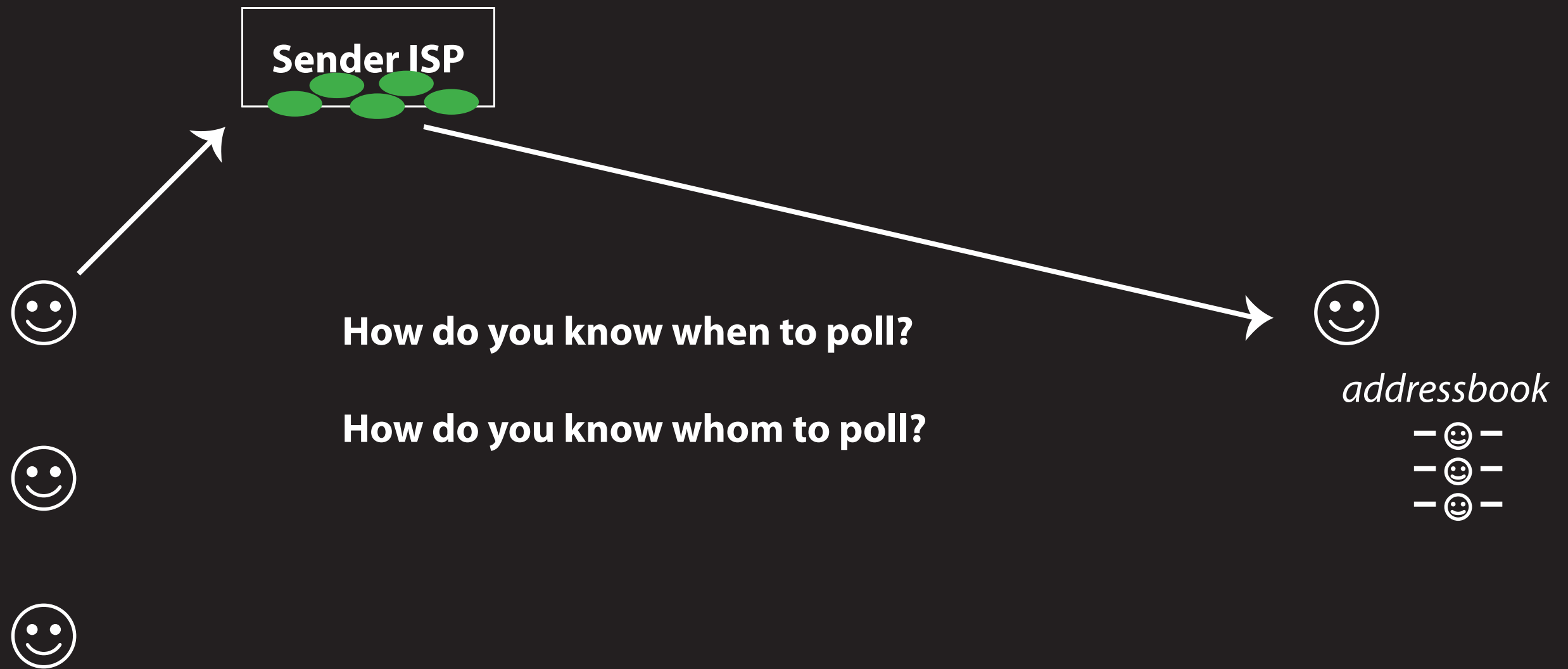
**Sender ISP**

**How do you know when to poll?**

**Sender ISP**

How do you know when to poll?

How do you know whom to poll?

*addressbook*

Sender ISP → Receiver ISP

How do you know when to poll?

*addressbook*

**Stubs:**
**Senders notify receivers with UDP.**
**"Come and get it, big boy!"**

mail from: sender1@sender-isp.com
rcpt to: receiver1@receiver-isp.com
url: http://sender-isp.com/sender1/receiver1/

Sender ISP

Sender ISP

Sender ISP

Receiver ISP

● in DB
vs
⬤ on disk

addressbook

**Stubs:**
**Senders notify receivers with UDP.**
**"Come and get it, big boy!"**

mail from: sender1@sender-isp.com
rcpt to: receiver1@receiver-isp.com
url: http://sender-isp.com/sender1/receiver1/

**Sender ISP**

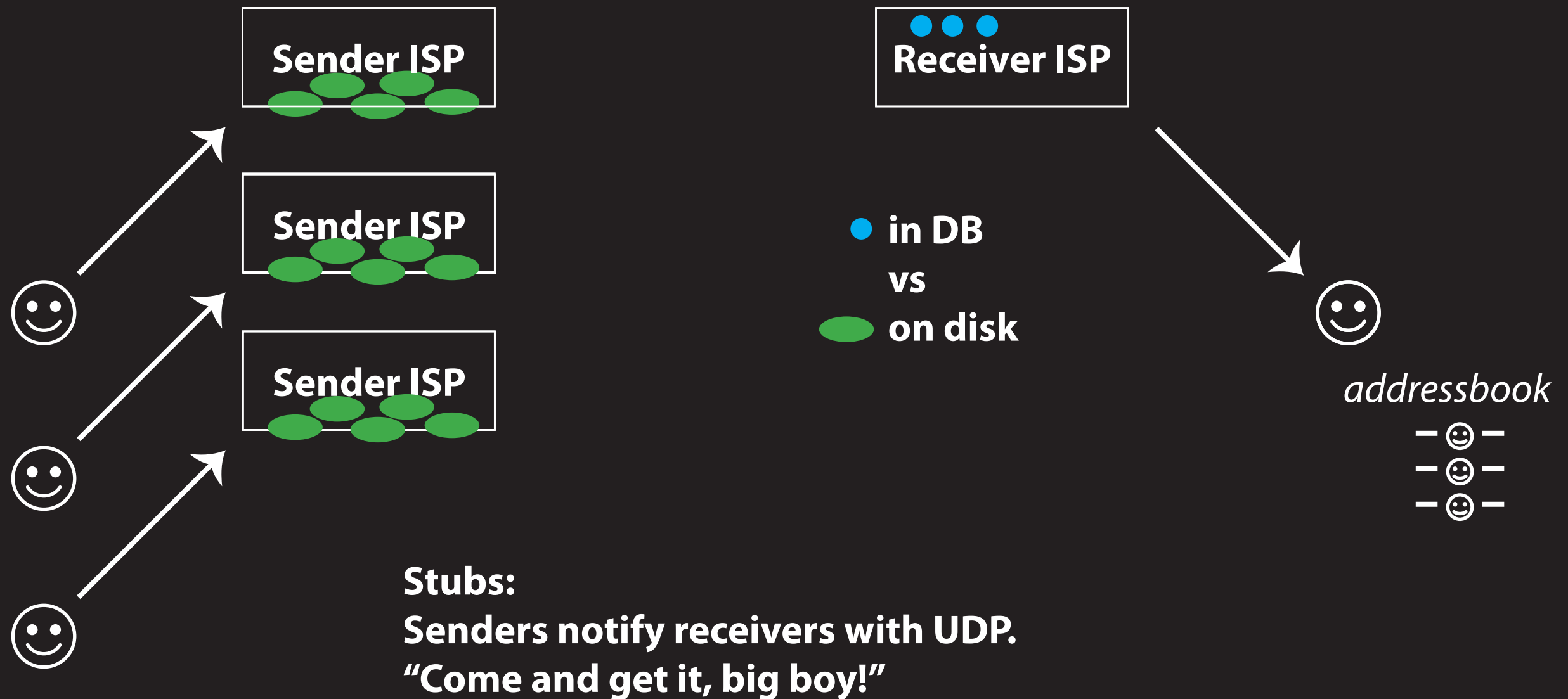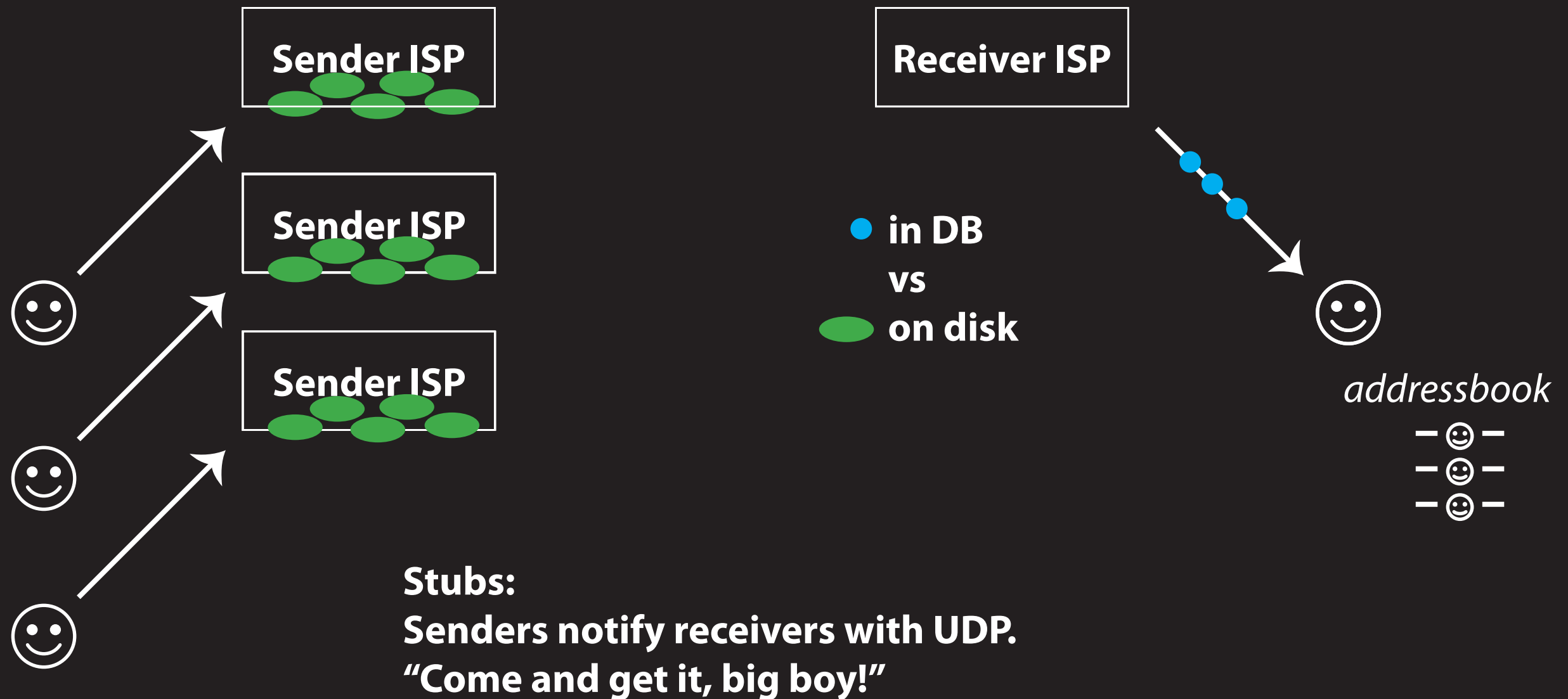**Sender ISP**

**Sender ISP**

**Receiver ISP**

● **in DB**
**vs**
● **on disk**

*addressbook*

**Stubs:**
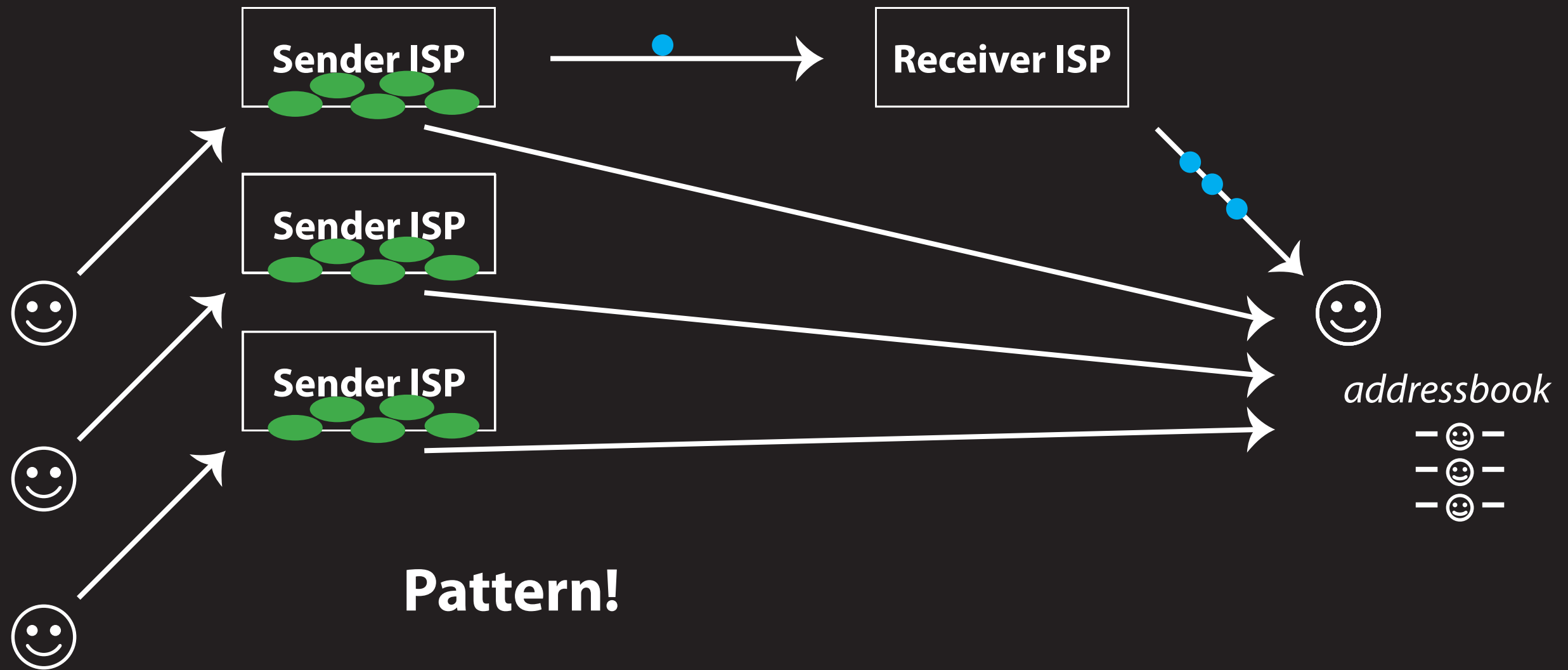**Senders notify receivers with UDP.**
**"Come and get it, big boy!"**

**mail from: sender1@sender-isp.com**
**rcpt to: receiver1@receiver-isp.com**
**url: http://sender-isp.com/sender1/receiver1/**

Sender ISP

Sender ISP

Sender ISP

Receiver ISP

addressbook

**Pattern!**

**Push the notifications.**
**Pull the content.**

# The First Contact Problem.

**The First Contact Problem.**

**(People add one another to IM buddy lists just fine.)**

# The First Contact Problem.

# Use Stubs + reputation *systems.*

# The First Contact Problem.

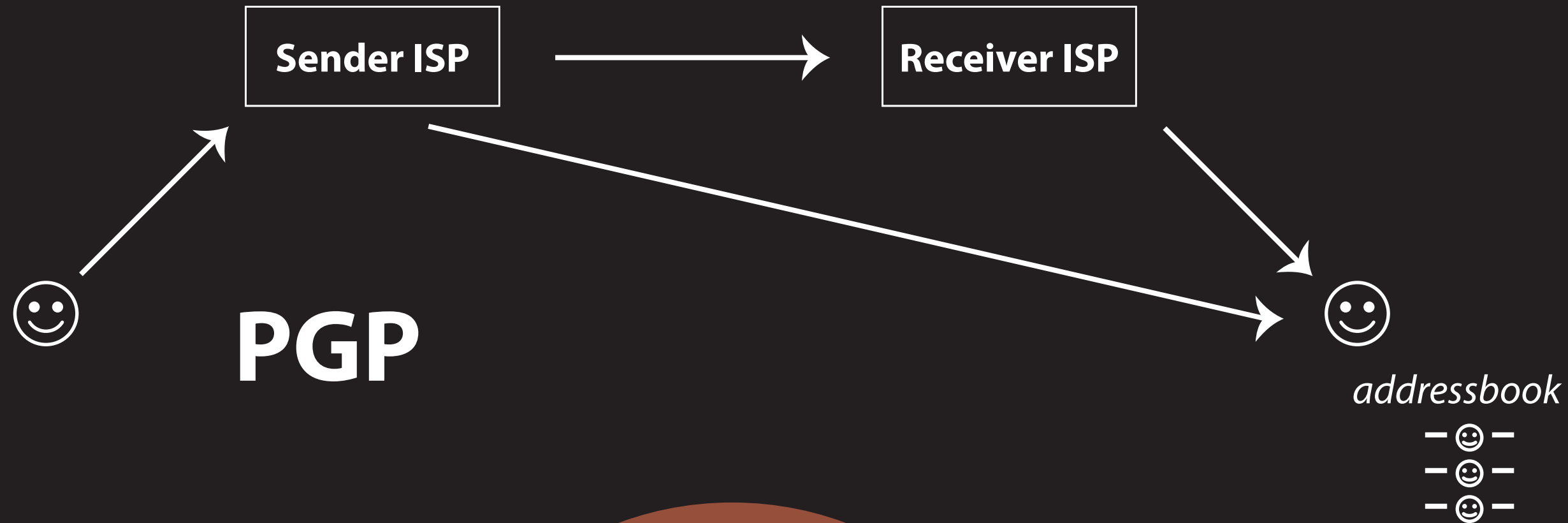## Here's my business card. Please mail me.

The opposite of every great idea is another great idea.
— Niels Bohr

# The First Contact Problem.

## Here's my business card. Please let me mail you.

*Simplest Possible Implementation*

**Sender ISP** → **Receiver ISP**

**PGP**

☺

*addressbook*
☺
☺
☺

**Our Problem**
- retrieve mail over HTTP GET
- PGP keyring contains the URLs
- SMTP proxy + submission CGI
- the sender MUA encrypts?
- receiver MTA or MUA decrypts.
- push notification with UDP.
- Receiver ISP stores "dirty list".
- Polling with UDP also.

**Out of Scope**

Lightweight UDP notification

discovery with _stub._udp.x.com

discovery with HTTP /$url/udp.xml

Use PGP for addressbook

Use some other addressbook

UDP polling

key parties

**Core**
**HTTP GET**
**Full Crypto**

webmail or blog-style submission

FOAF-style pubring.gpg

encrytion by the MUA? by the server?

smtp2stub + post_manager.cgi

*Simplest Possible Implementation*

**Sender ISP** → **Receiver ISP**

☺

# FOAF

*addressbook*

☺

**Our Problem**
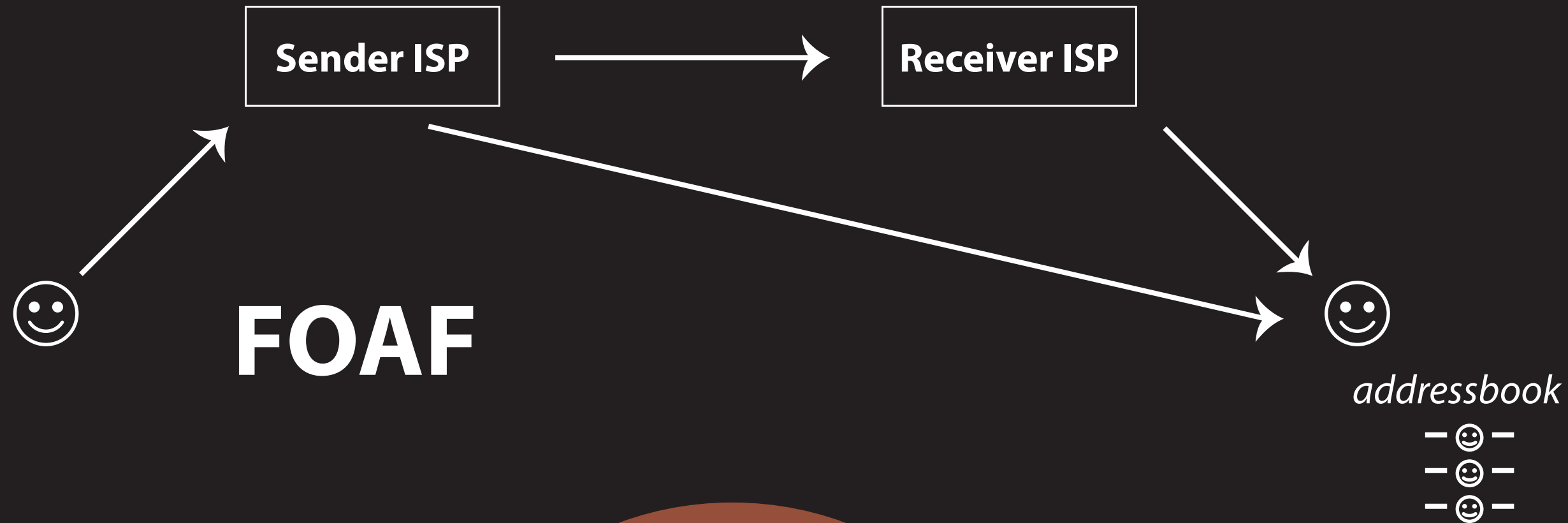- retrieve mail over HTTP GET
- PGP keyring contains the URLs
- SMTP proxy + submission CGI
- the sender MUA encrypts?
- receiver MTA or MUA decrypts.
- push notification with UDP.
- Receiver ISP stores "dirty list".
- Polling with UDP also.

**Out of Scope**

Lightweight UDP notification

discovery with _stub._udp.x.com

discovery with HTTP /$url/udp.xml

Use PGP for addressbook

Use some other addressbook

UDP polling

key parties

**Core**
HTTP GET
Full Crypto

webmail or blog-style submission

FOAF-style pubring.gpg

encrytion by the MUA? by the server?

smtp2stub + post_manager.cgi

*Simplest Possible Implementation*

**Sender ISP** ➔ **Receiver ISP**

☺

# Social Networks

☺

*addressbook*

─ ☺ ─
─ ☺ ─
─ ☺ ─

**Our Problem**
- retrieve mail over HTTP GET
- PGP keyring contains the URLs
- SMTP proxy + submission CGI
- the sender MUA encrypts?
- receiver MTA or MUA decrypts.
- push notification with UDP.
- Receiver ISP stores "dirty list".
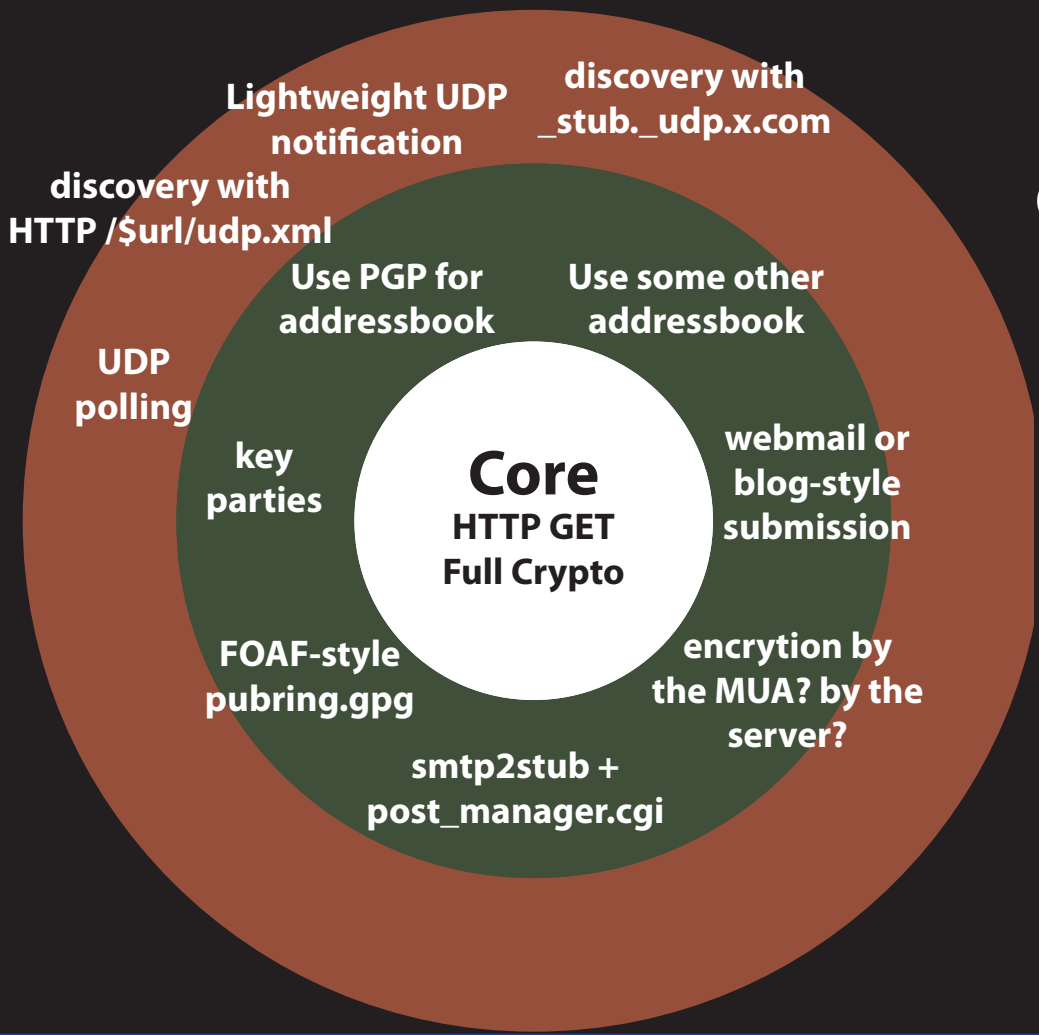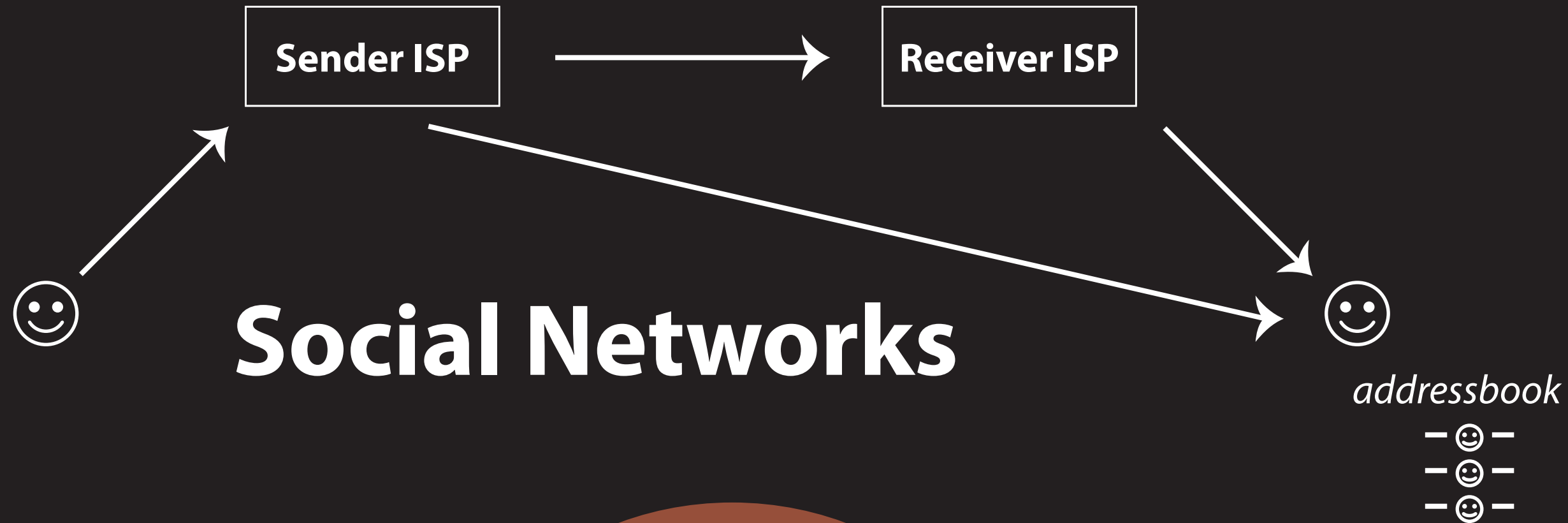- Polling with UDP also.

**Out of Scope**

Lightweight UDP
notification

discovery with
_stub._udp.x.com

discovery with
HTTP /$url/udp.xml

Use PGP for
addressbook

Use some other
addressbook

UDP
polling

key
parties

**Core**
HTTP GET
Full Crypto

webmail or
blog-style
submission

FOAF-style
pubring.gpg

encrytion by
the MUA? by the
server?

smtp2stub +
post_manager.cgi

*Simplest Possible Implementation*

**Sender ISP** → **Receiver ISP**

☺

# Email 2.0!

☺

*addressbook*
–☺–
–☺–
–☺–

**Our Problem**
- retrieve mail over HTTP GET
- PGP keyring contains the URLs
- SMTP proxy + submission CGI
- the sender MUA encrypts?
- receiver MTA or MUA decrypts.
- push notification with UDP.
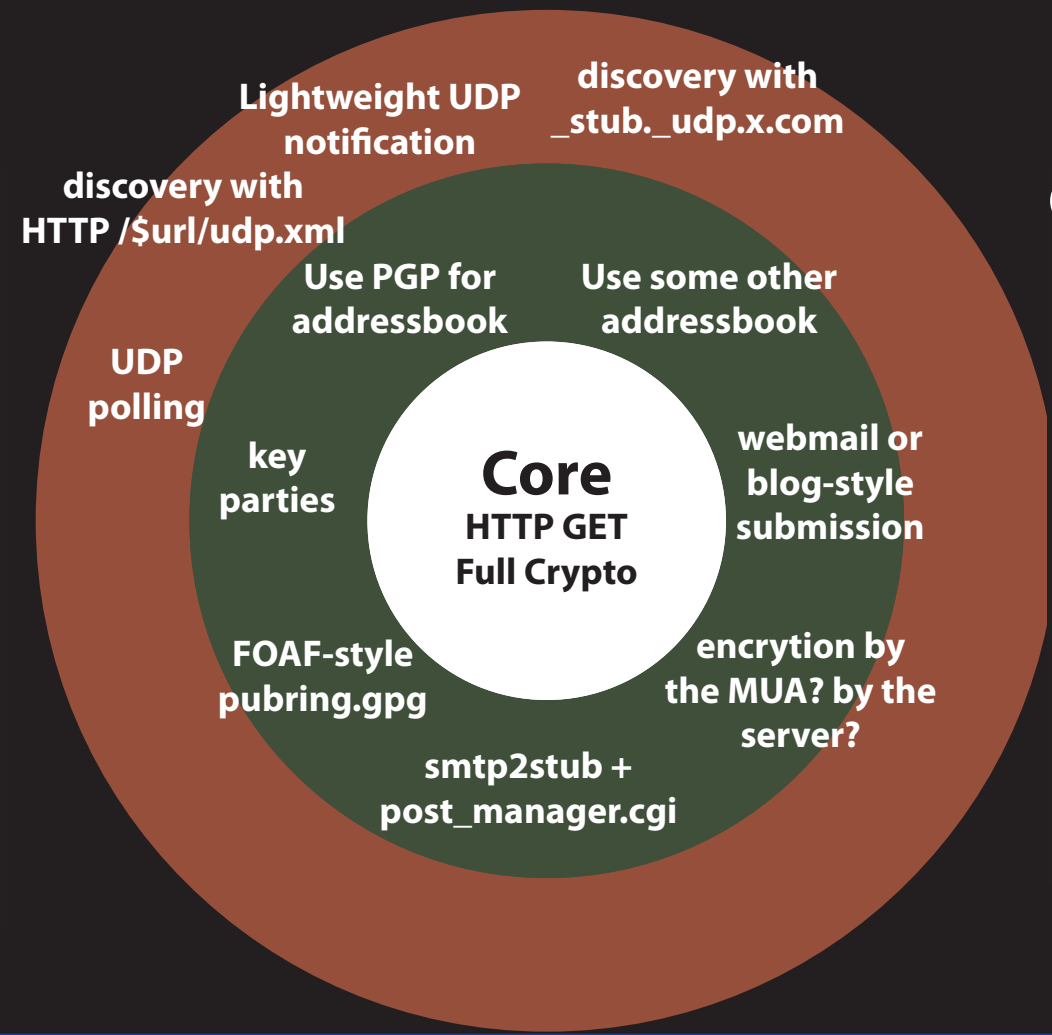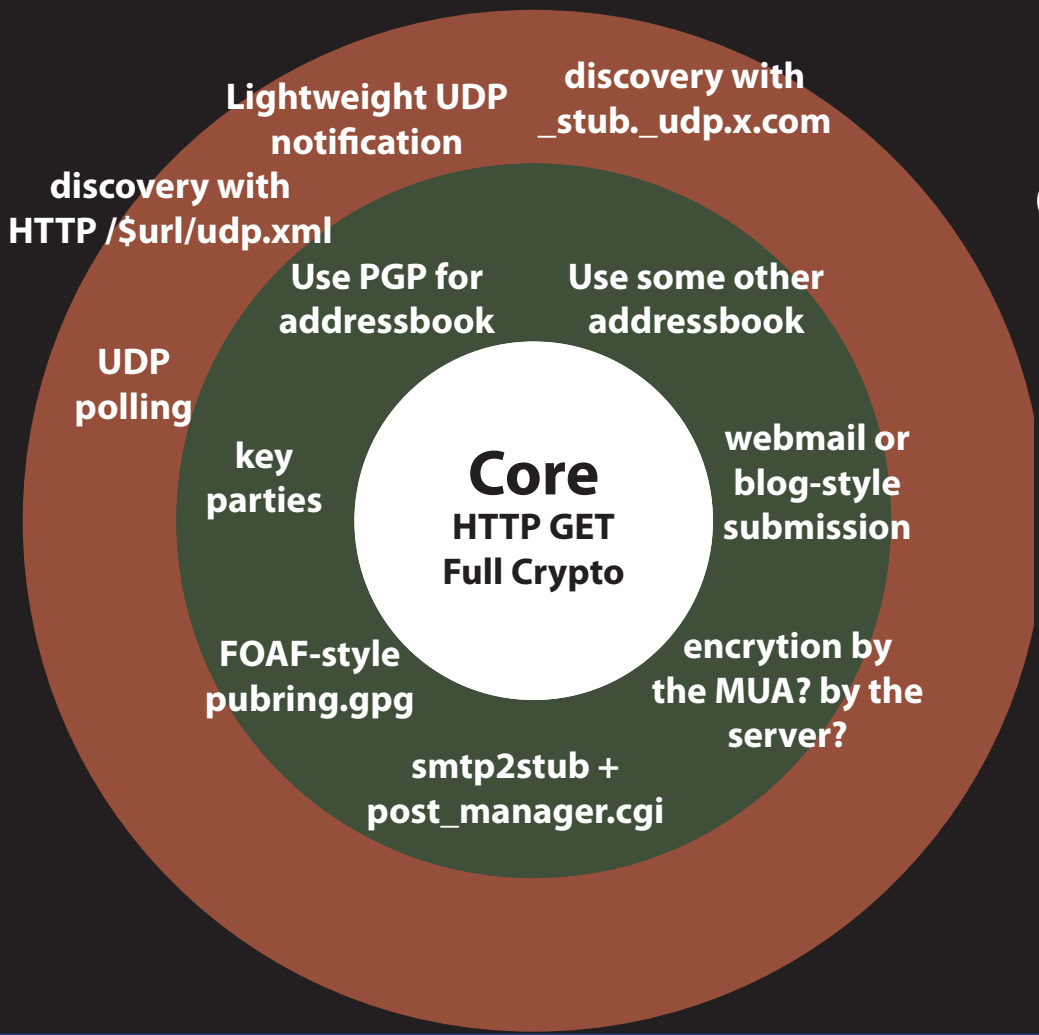- Receiver ISP stores "dirty list".
- Polling with UDP also.

**Out of Scope**

Lightweight UDP notification

discovery with _stub._udp.x.com

discovery with HTTP /$url/udp.xml

Use PGP for addressbook

Use some other addressbook

UDP polling

key parties

**Core**
HTTP GET
Full Crypto

webmail or blog-style submission

FOAF-style pubring.gpg

encrytion by the MUA? by the server?

smtp2stub + post_manager.cgi

**July 19 2006**
**Tech Talk at Google,**
**available on Google Video**

**July 26 2006**
**lightning talk at OScon**

# http://stubmail.com

# Volunteers welcome.

*Thanks to Nathan Cheng, Richard Soderberg, Andrei Freeman, Ali Farschian,
Nat Torkington, Mark-Jason Dominus, Andy Newton, djb*

*Produced in Adobe InDesign CS2*