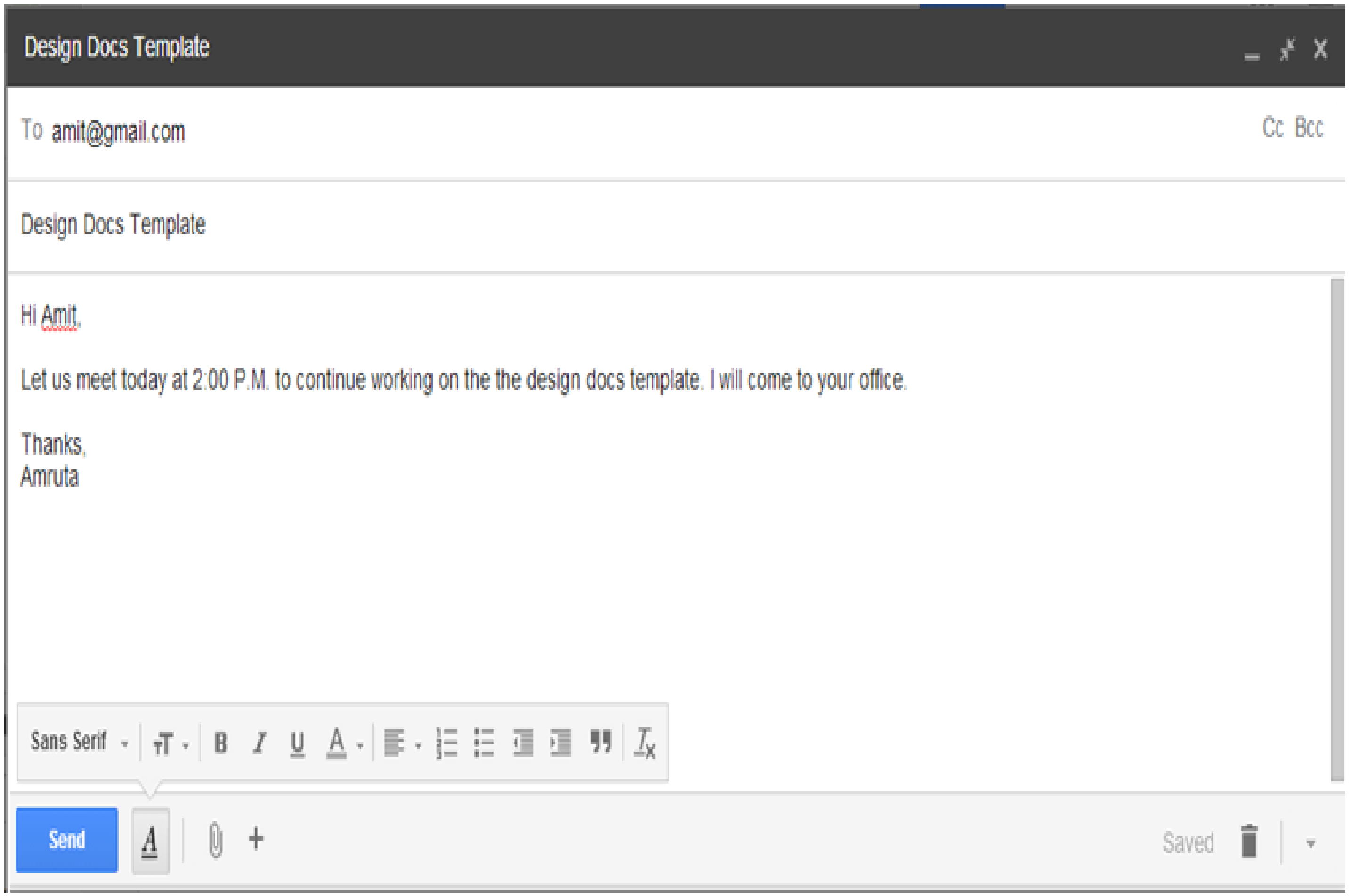
Moving from User Documentation to Developer Documentation

Amruta Ranade

druva

What are Developer Docs

User Docs Vs. Dev Docs



Design Docs Template

To amit@gmail.com

Cc Bcc

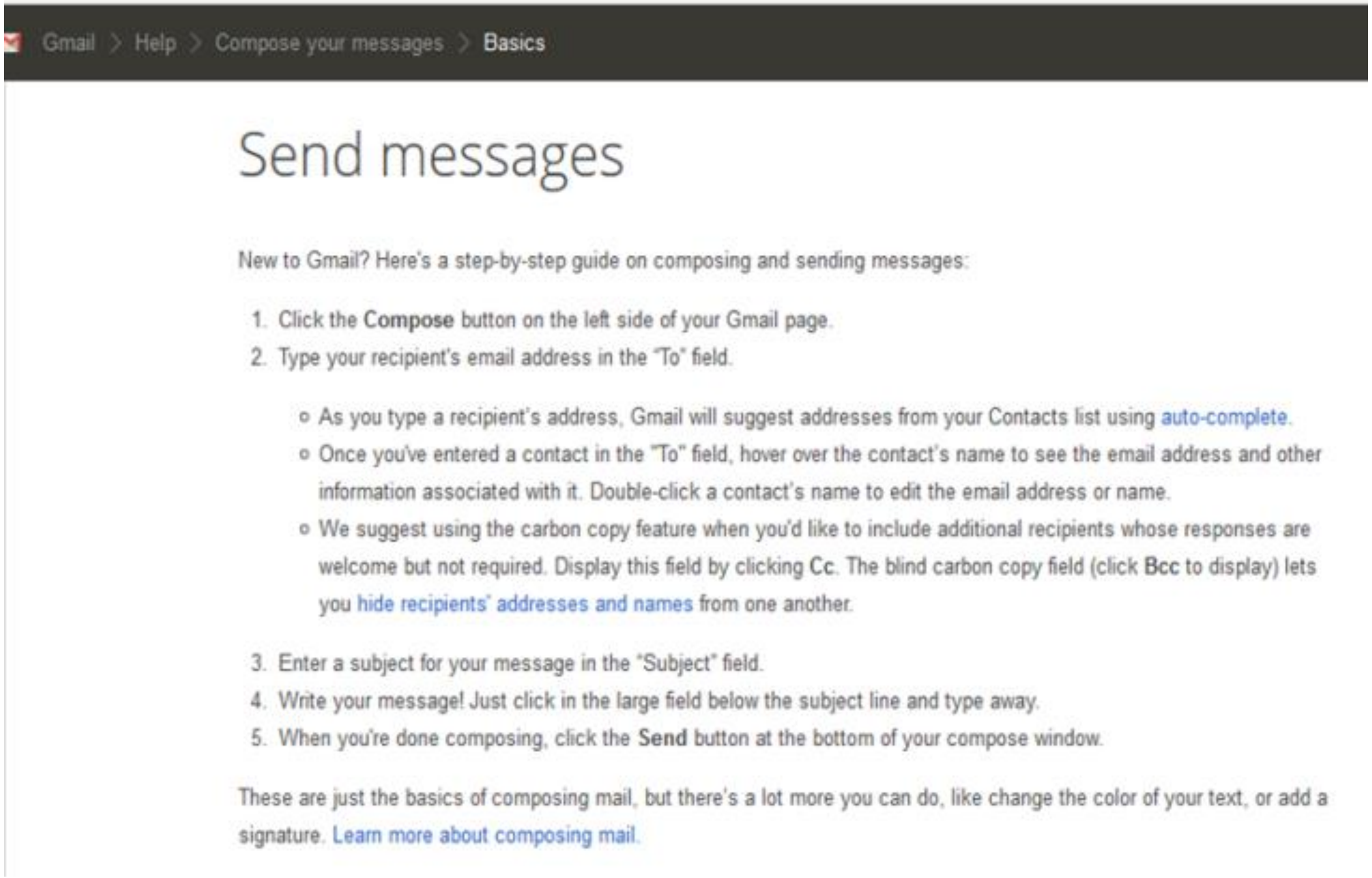
Design Docs Template

Hi Amit,

Let us meet today at 2:00 P.M. to continue working on the the design docs template. I will come to your office.

Thanks,
Amruta

User documentation



Send messages

New to Gmail? Here's a step-by-step guide on composing and sending messages:

1. Click the **Compose** button on the left side of your Gmail page.
2. Type your recipient's email address in the "To" field.
    - As you type a recipient's address, Gmail will suggest addresses from your Contacts list using auto-complete.
    - Once you've entered a contact in the "To" field, hover over the contact's name to see the email address and other information associated with it. Double-click a contact's name to edit the email address or name.
    - We suggest using the carbon copy feature when you'd like to include additional recipients whose responses are welcome but not required. Display this field by clicking **Cc**. The blind carbon copy field (click **Bcc** to display) lets you hide recipients' addresses and names from one another.
3. Enter a subject for your message in the "Subject" field.
4. Write your message! Just click in the large field below the subject line and type away.
5. When you're done composing, click the **Send** button at the bottom of your compose window.

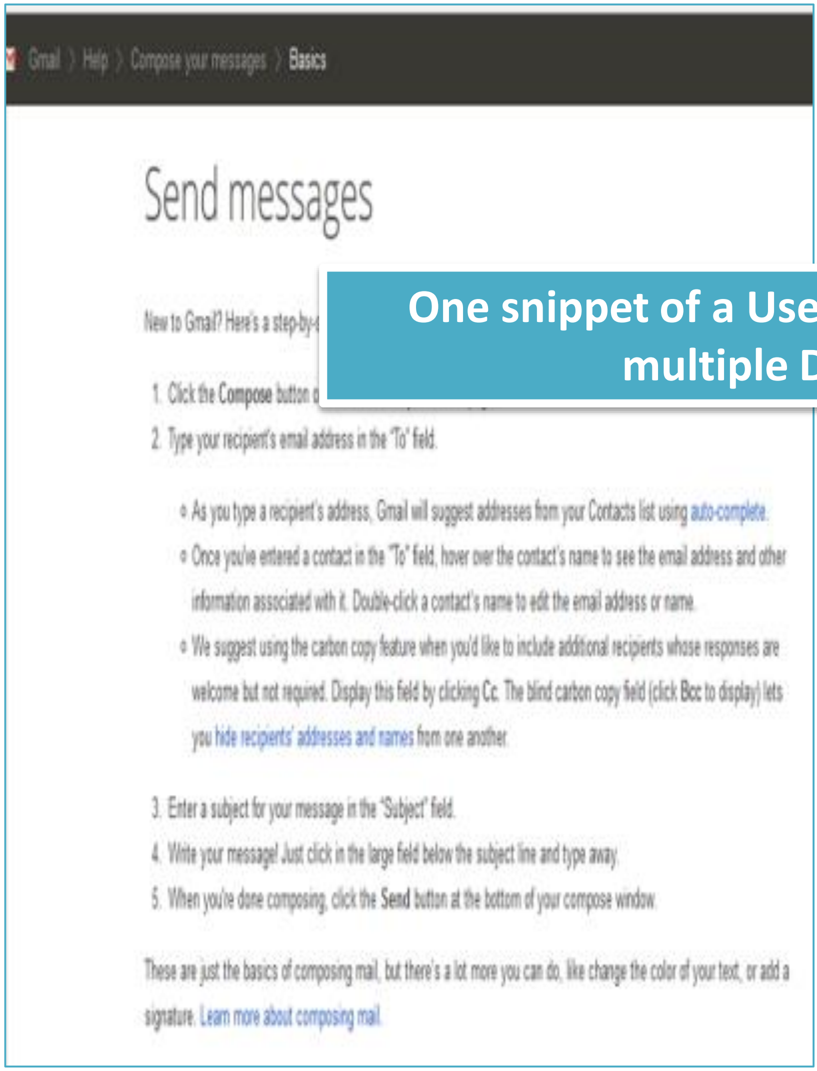
These are just the basics of composing mail, but there's a lot more you can do, like change the color of your text, or add a signature. Learn more about composing mail.

Developer documentation

- What is the format of the email?
- How is the message encapsulated to form the message packet? Description of the headers and metadata
- How is the message packet converted into a bit stream?
- How does the Client notify the Server that it wants to send an email
- How does the Client connect to the Server?
- Which protocol is used?
- How does the Server look up the recipient's address?
- How does the Server figure out the recipient’s Email Server?
- ….

User Docs

Dev Docs



Gmail > Help > Compose your messages > Basics

Send messages

New to Gmail? Here's a step-by-s

1. Click the **Compose** button o
2. Type your recipient's email address in the "To" field.
   - As you type a recipient's address, Gmail will suggest addresses from your Contacts list using auto-complete.
   - Once you've entered a contact in the "To" field, hover over the contact's name to see the email address and other information associated with it. Double-click a contact's name to edit the email address or name.
   - We suggest using the carbon copy feature when you'd like to include additional recipients whose responses are welcome but not required. Display this field by clicking Cc. The blind carbon copy field (click **Bcc** to display) lets you hide recipients' addresses and names from one another.
3. Enter a subject for your message in the "Subject" field.
4. Write your message! Just click in the large field below the subject line and type away.
5. When you're done composing, click the **Send** button at the bottom of your compose window.

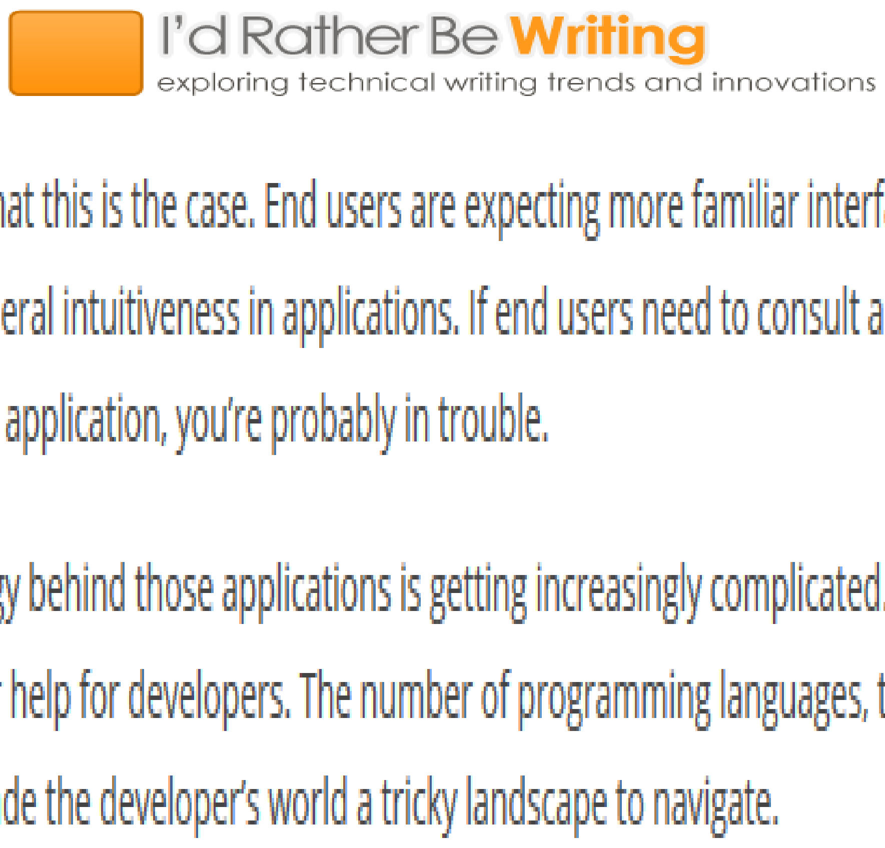
These are just the basics of composing mail, but there's a lot more you can do, like change the color of your text, or add a signature. Learn more about composing mail.

- What is the format of the email?
- How is the message encapsulated to form the message packet? Description of the headers and metadata
- ket
  m?
- How does the Client notify the Server that it wants to send an email
- How does the Client connect to the Server?
- Which protocol is used?
- How does the Server look up the recipient's address?
- How does the Server figure out the recipient's Email Server?

One snippet of a User document corresponds to multiple Dev documents ..

druva

Why do we need Dev Docs?



I'd Rather Be **Writing**
exploring technical writing trends and innovations

Andrew agreed that this is the case. End users are expecting more familiar interfaces, increased usability, and general intuitiveness in applications. If end users need to consult a help file to use your mainstream application, you're probably in trouble.

But the technology behind those applications is getting increasingly complicated. More than ever there's a need for help for developers. The number of programming languages, technologies, and platforms has made the developer's world a tricky landscape to navigate.

User Docs

Resources: One resource to document a feature

When: After the feature is designed

Written for:

- End-users
- Support

Dev Docs

Resources: More than one resource depending on the technical depth of the feature

When: Before the feature is designed

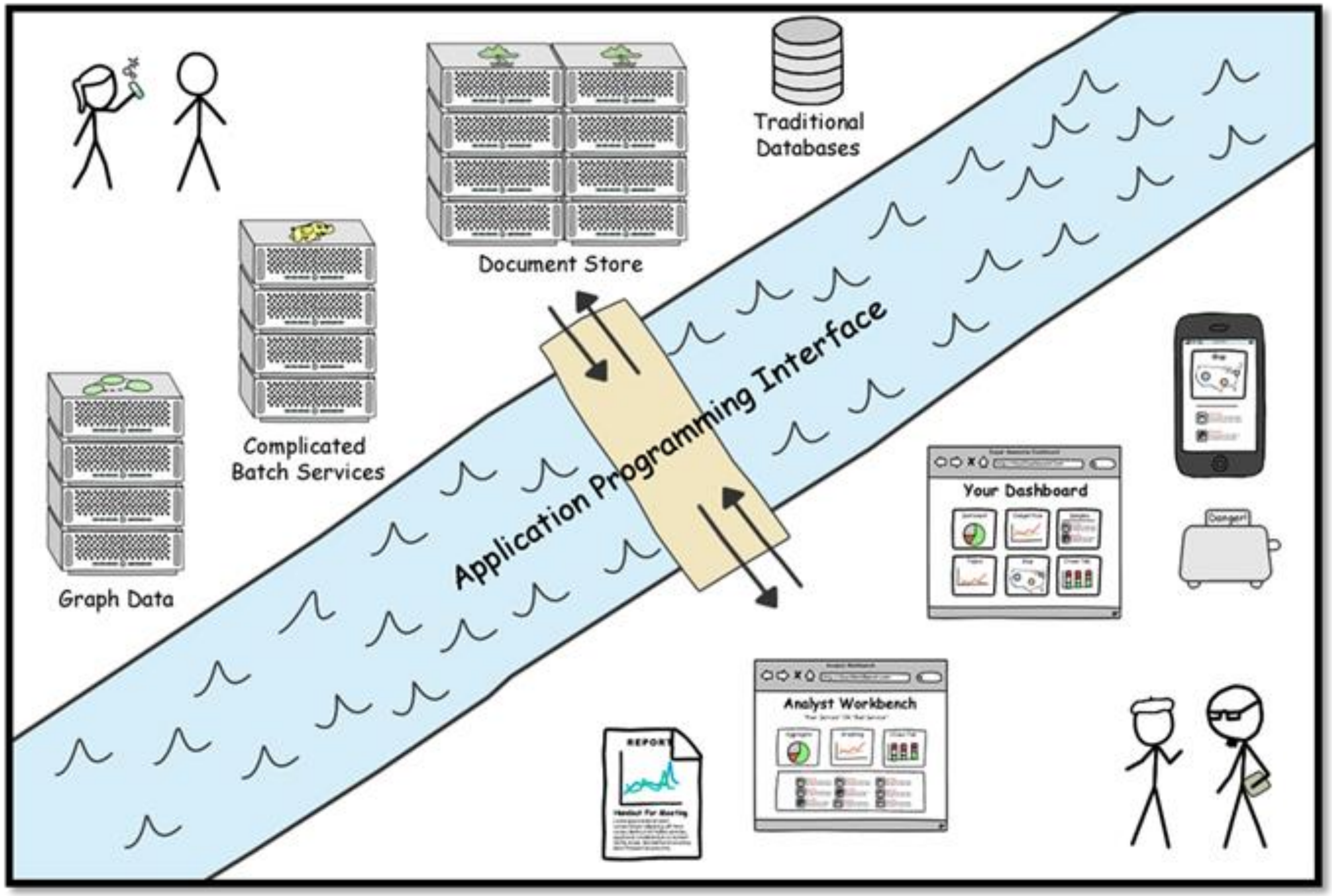
Written for:

- Developers
- QA
- Tech Writing
- Support
- Sales and Marketing
- Project Management

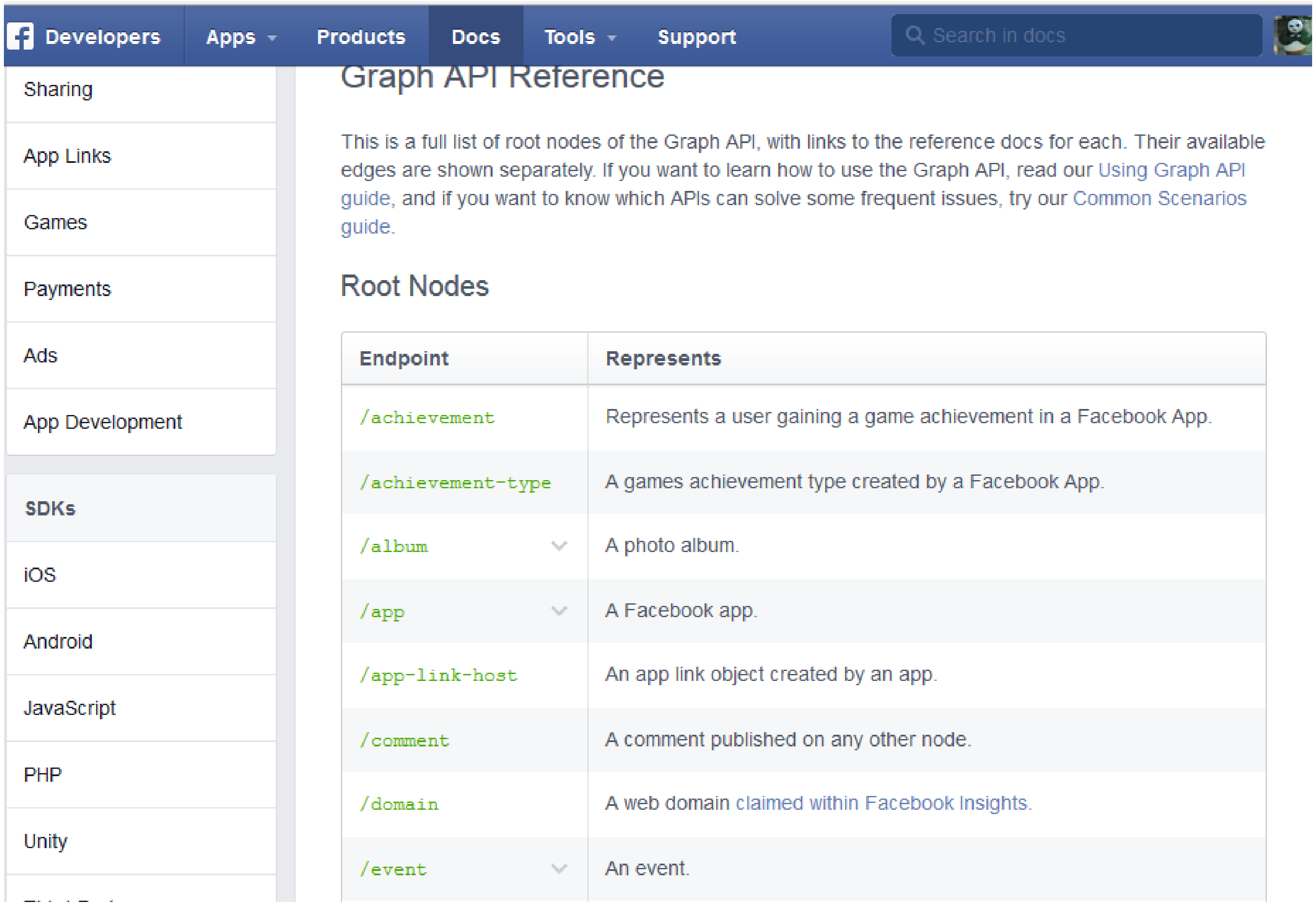
Types of Dev Docs

API Docs

What is an API



Example of an API Doc



Graph API Reference

This is a full list of root nodes of the Graph API, with links to the reference docs for each. Their available edges are shown separately. If you want to learn how to use the Graph API, read our Using Graph API guide, and if you want to know which APIs can solve some frequent issues, try our Common Scenarios guide.

Root Nodes

Endpoint	Represents
`/achievement`	Represents a user gaining a game achievement in a Facebook App.
`/achievement-type`	A games achievement type created by a Facebook App.
`/album`	A photo album.
`/app`	A Facebook app.
`/app-link-host`	An app link object created by an app.
`/comment`	A comment published on any other node.
`/domain`	A web domain claimed within Facebook Insights.
`/event`	An event.

Components of an API Doc

- Overview:

  Explain what advantages developers have in using the platform, and in some cases, provide an architectural description of the platform.
- Getting started:

  Help the developer get started, in the form of step-by-step tutorials or simpler walkthroughs.
- Sample code:

  Provide well-commented code samples that developers can build on.
- Reference material:

  Provide detailed information about each class, member, function or XML element.

Resources

- Tom Johnson – http://idratherbewriting.com/
- Sarah Maddox – API Technical Writing http://www.slideshare.net/sarahmaddox/api-technical-writing
- A Coder's Guide to Writing API Documentation – http://msdn.microsoft.com/en-us/magazine/gg309172.aspx

Software Architecture Docs

What is Software Architecture

“Software architecture is the software’s fundamental organization, embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.”

How do you document a Software Architecture

Software architecture is documented in terms of views.

To understand views, consider an architect designing a building…

druva

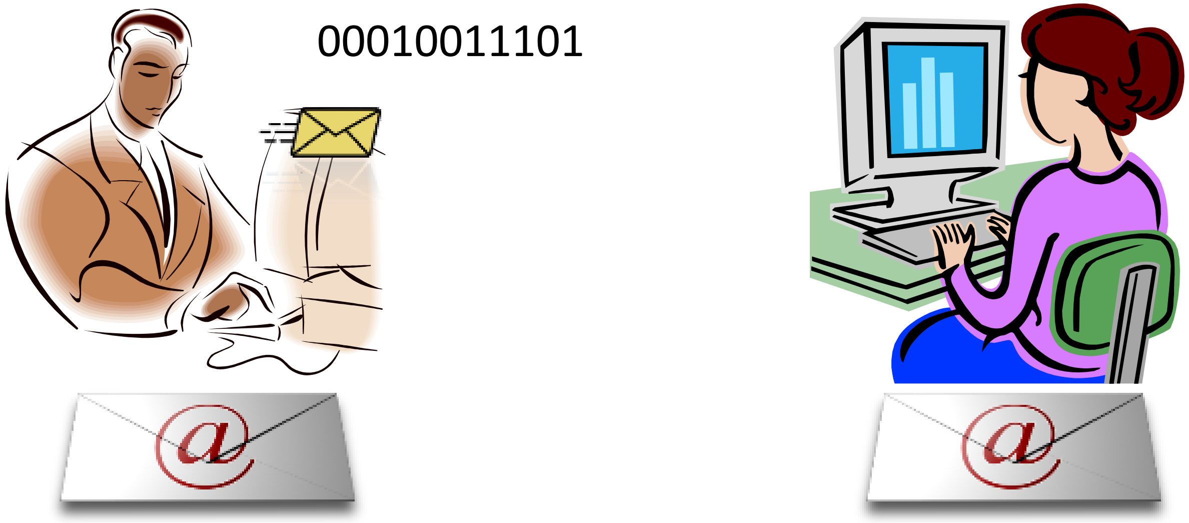
Understanding Views

Just as a building architect might create wiring diagrams, floor plans, and elevations to describe different facets of a building to its different stakeholders (electricians, owners, planning officials), so an IT architect might create physical and security views of an IT system for the stakeholders who have concerns related to these aspects.

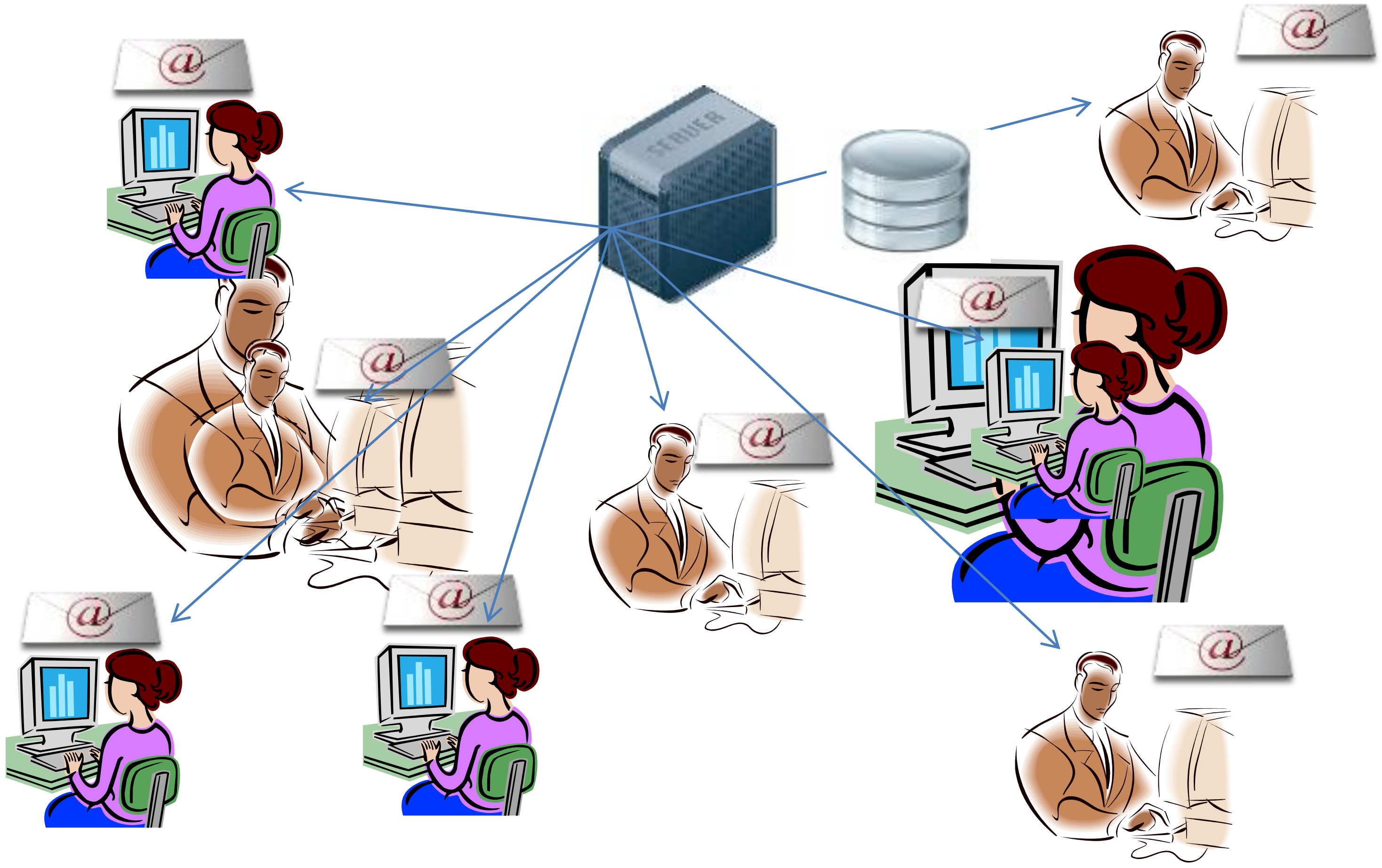
Context View

Start from scratch and build the story of the evolution of the product to the Big Picture

Context View



Context View



Functional and Non-functional View

What does the software do?

Functional View

Key functions of the system:

- Activation
- Authentication
- Configuration
- Send and receive emails
- Send attachments
- Send emails to multiple users
- Chat
- Video Chat
- Group Chat
- Send email on behalf of someone
- Signature
- Encryption
- Folder structure

Non-functional View

Examples:

- Performance (latency and throughput)
- Scalability (data and traffic volumes)
- Availability (uptime, downtime, scheduled maintenance, 24x7)
- Security (authentication, authorisation, data confidentiality)
- Flexibility
- Monitoring and management
- Reliability
- Accessibility

Logical View

Drill down to the components and connections

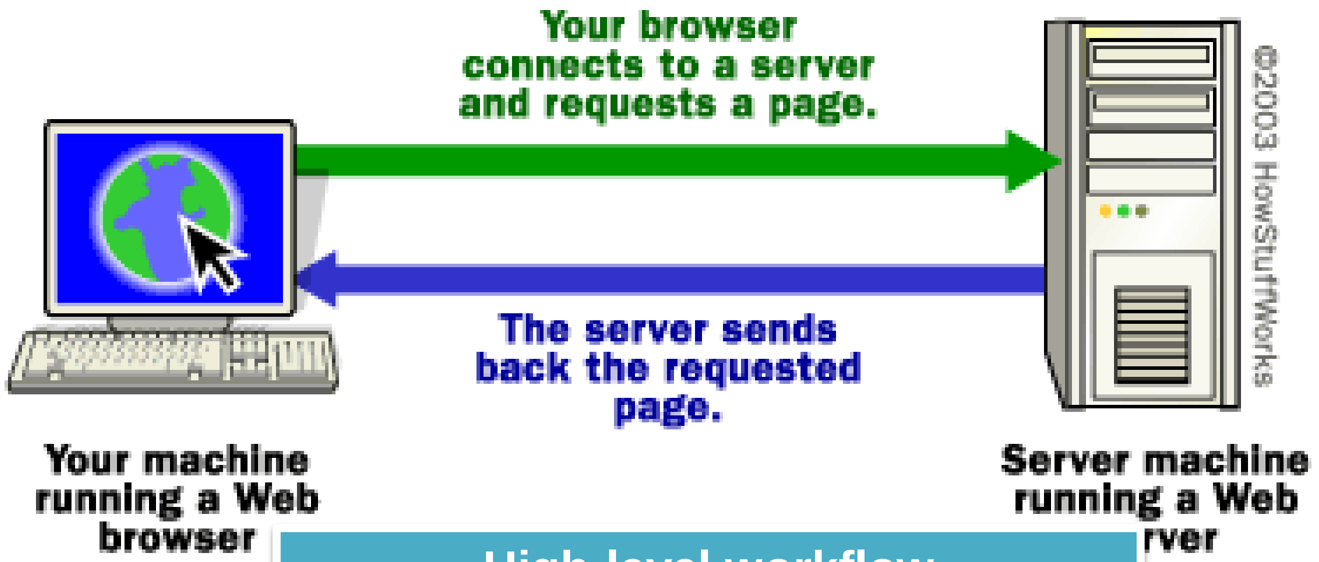
Components

- Message
  - Composition
  - Headers
  - Metadata
- Email Client
  - Component to format the message
  - Component to submit message to server
  - Component to encrypt the message
- Outgoing Mail Server
  - SMTP Server
- Incoming Mail Server
  - POP3 Server
- Protocol
  - HTTP
- Email Storage
- Contact Database
- Ports

Process View

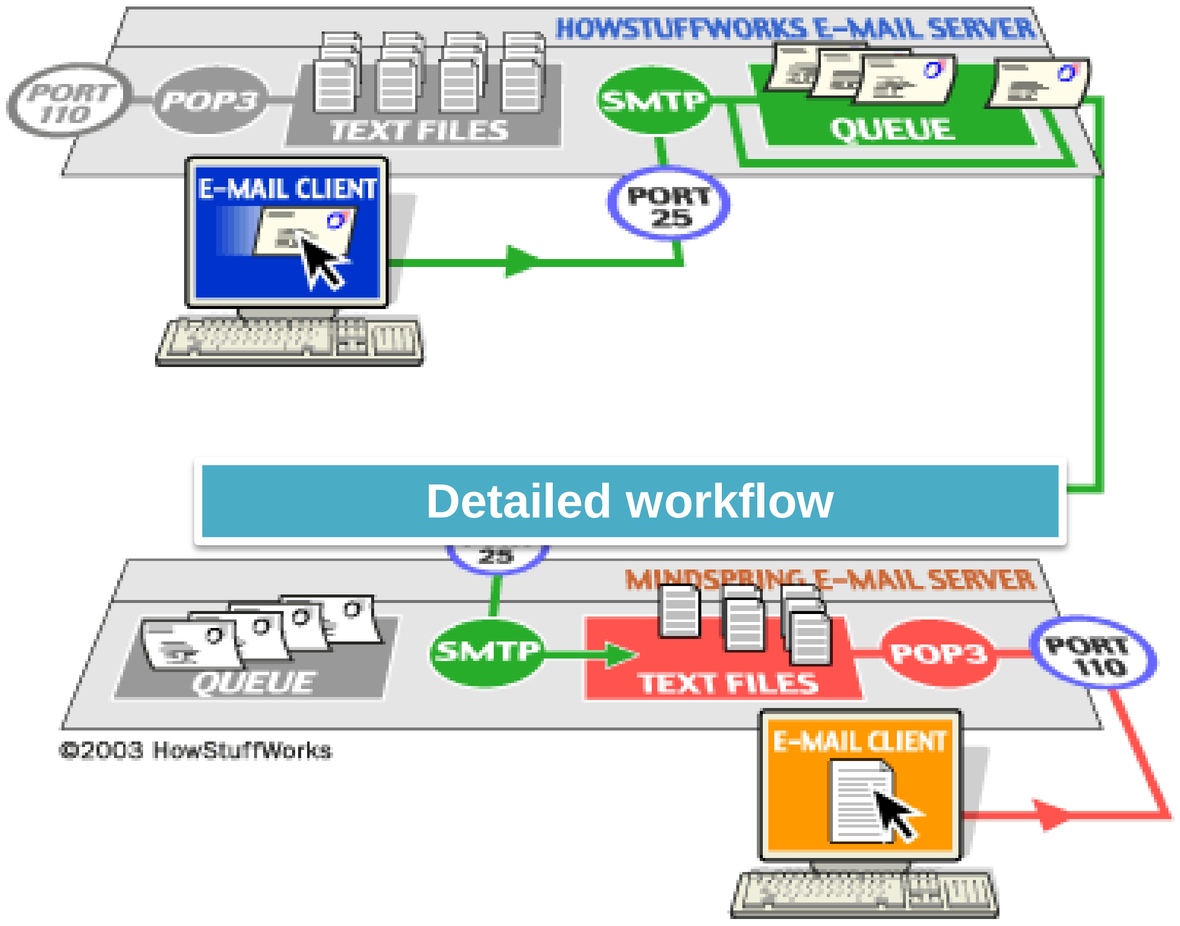
Workflow for each task

Process View


Your browser connects to a server and requests a page.
The server sends back the requested page.
Your machine running a Web browser
Server machine running a Web server
©2003 HowStuffWorks

High-level workflow

Process View



Detailed workflow

Design View

Define the details of each component

Design View

Email Client

- **Message composition**

  Email clients usually contain user interfaces to display and edit text. Some applications permit the use of a program-external editor.

  The email clients will perform formatting according to RFC 5322 for headers and body, and MIME for non-textual content and attachments. Headers include the destination fields, *To*, *Cc*, and *Bcc*, and the originator fields *From* which is the message's author(s), *Sender* in case there are more authors, and *Reply-To* in case responses should be addressed to a different mailbox. To better assist the user with destination fields, many clients maintain one or more address books and/or are able to connect to an LDAP directory server. For originator fields, clients may support different identities.

  Client settings require the user's *real name* and *email address* for each user's identity, and possibly a list of LDAP servers.

Interface View

How do the components communicate

Interface View

SMTP

- SMTP stands for Simple Mail Transfer Protocol. The SMTP server handles outgoing mail.
- The SMTP server listens on well-known port number 25, POP3 listens on port 110 and IMAP uses port 143
- Whenever you send a piece of e-mail, your e-mail client interacts with the SMTP server to handle the sending. The SMTP server on your host may have conversations with other SMTP servers to deliver the e-mail.
- **Commands:**
  - **HELO** - introduce yourself
  - **EHLO** - introduce yourself and request extended mode
  - **MAIL FROM:** - specify the sender
  - **RCPT TO:** - specify the recipient
  - **DATA** - specify the body of the message (To, From and Subject should be the first three lines.)
  - **RSET** - reset

Interface View

POP3

- In the simplest implementations of POP3, the server really does maintain a collection of text files -- one for each e-mail account. When a message arrives, the POP3 server simply appends it to the bottom of the recipient's file.
- When you check your e-mail, your e-mail client connects to the POP3 server using port 110. The POP3 server requires an account name and a password. Once you've logged in, the POP3 server opens your text file and allows you to access it.
- **Commands:**
  - **PASS** - enter your password
  - **QUIT** - quit the POP3 server
  - **LIST** - list the messages and their size
  - **RETR** - retrieve a message, pass it a message number
  - **DELE** - delete a message, pass it a message number
  - **TOP** - show the top x lines of a message, pass it a message number and the number of lines

Technology Selection

Why did you choose X?

POP3

- **Workflow:**
  - Connect to server
  - Retrieve all mail
  - Store locally as new mail
  - Delete mail from server
  - Disconnect
- **Advantages:**
  - Mail stored locally, always accessible, even without internet connection
  - Internet connection needed only for sending and receiving mail
  - Saves server storage space
  - Option to leave copy of mail on server
  - Consolidate multiple email accounts and servers into one inbox
- **Choose POP if…**
  - You want to access your mail from only one single device
  - You need constant access to your email, regardless of internet availability
  - Your server storage space is limited

IMAP

- **Workflow:**
  - Connect to server
  - Fetch user requested content and cache it locally, Process user edits
  - Disconnect
- **Advantages:**
  - Mail stored on remote server, i.e. accessible from multiple different locations
  - Internet connection needed to access mail
  - Faster overview as only headers are downloaded until content is explicitly requested
  - Mail is automatically backed up if server is managed properly
  - Saves local storage space
  - Option to store mail locally
- **Choose IMAP if…**
  - You want to access your email from multiple different devices
  - You have a reliable and constant internet connection
  - You want to receive a quick overview of new emails or emails on the server
  - Your local storage space is limited

Other Views

- Data View
- Implementation View
- Deployment View
- Operational View
- Infrastructure View

You don’t have to document all the views. Pick and choose the views that are applicable to you, depending on the audience.

For instance, the Deployment team will be interested in the Deployment and Infrastructure views, while the Database Admin will be interested in the Data View.

Design Docs

Design Docs

- Software Architecture docs are system-level docs. They are the base of Dev docs.

  Design docs are feature specific. They assume the knowledge covered in the software architecture docs and build on it. They are incremental additions to the architecture docs.

- Design docs usually accompany an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design.

Design Docs

What should a Design Doc include:

- Overview/Concept/Problem statement
- Context diagram and background information
- Possible solutions
- Analysis of possible solutions
- Solution chosen and rationale
- Areas impacted by chosen solution
- Workflow
- Code implementation
- Scenarios not addressed
- References

Design Doc Example

- **Requirement:**
  - User should be able to send and receive emails using Gmail.

- **Background and prerequisites:**

  Before understanding how Gmail sends and receives email, we need to understand:
  - Architecture of Gmail
  - How is an email composed
  - How SMTP works
  - How POP3 works

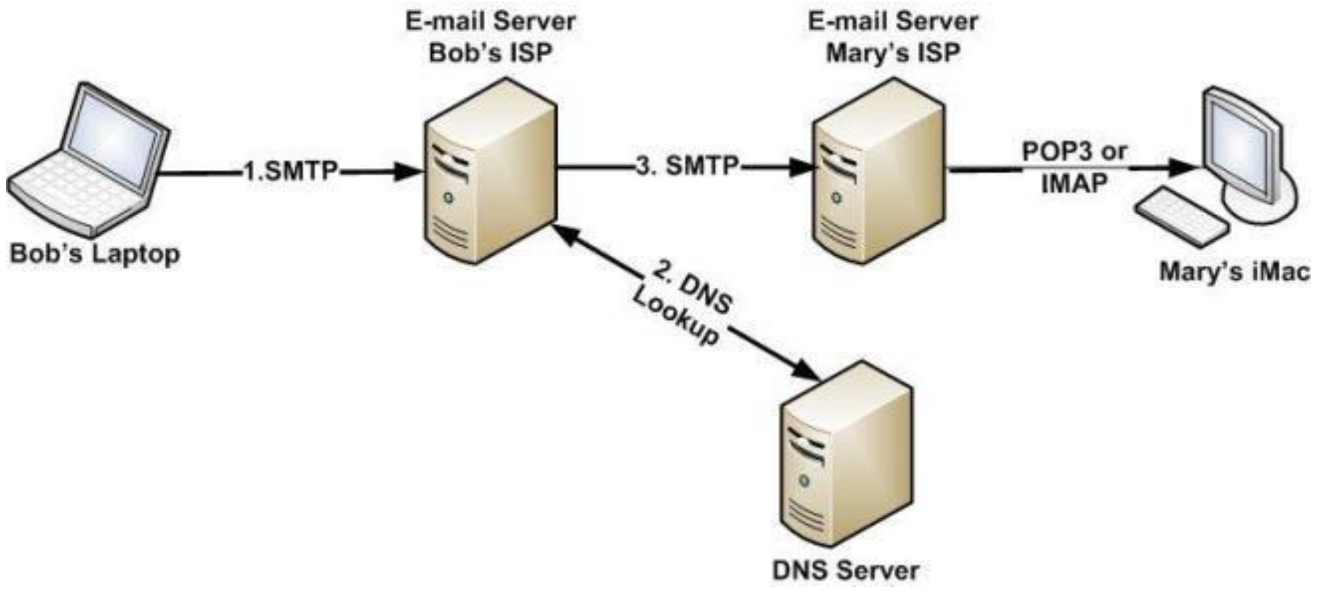
Design Doc Example

Solution:

1. User composes an email on the client.
2. Client connects to the server.
3. Server looks up the contact database and locates the recipient server.
4. Server transfers the email to the recipient server using the SMTP protocol.
5. Recipient server sends a notification to the recipient client that an email has been received.
6. Recipient client connects to the recipient server.
7. Email is transferred to the recipient client using the POP3 protocol.
8. Recipient client downloads the email and displays it to the user.

Design Doc Example

Solution:



Design Doc Example

Code implementation

- <Written by Developers>

Areas of impact

- Client
  - *Details*
- Server
  - *Details*

Scenarios not addressed

- Sending emails to multiple users has not been addressed in this solution.

Process

Software Architecture Docs

- Refer to User docs, sales and support, marketing docs
- Browse code
- Daily TOI sessions with Developers

Design Docs

- Sprint planning
- Team discussion meeting
- Engineering requirements meeting
- Design Review meeting
- Code review
- QA TOI/Demo
- Sales and Support TOI

Add links to user docs and sales and support, and marketing docs.

FAQs

These docs are very technical. Shouldn’t the Developers be writing these?

If developers want to say:

Miss Piggy wants to go to lunch, so she starts a new email and types Kermit's address in the "To:" box.

They end up writing:

Assume a function AddressOf(x) which is defined as the mapping from a user x, to the RFC-822 compliant email address of that user, an ANSI string. Let us assume user A and user B, where A wants to send an email to user B. So user A initiates a new message using any (but not all) of the techniques defined elsewhere, and types AddressOf(B) in the To: editbox.

FAQs

What are the challenges?

- Very technical
- Input to output ratio is skewed
- Need to use all skills of tech writing

What are the rewards?

- Real-time user feedback
- Sound technical knowledge – develop your own process to ramp up faster
- Niche field

FAQs

What are the skills of user documentation that can be applicable for developer documentation?

- Technical + Writing
- Analytical thinking
- Videos, Concept topics, Information Architecture
- Most important: Interviewing the SMEs

amruta2799@gmail.com
@AmrutaRanade