**How does gRPC work?**

|  |  |
| --- | --- |
| Diagram  Description automatically generated | RPC (Remote Procedure Call) is called “remote” because it enables communications between remote services when services are deployed to different servers under microservice architecture. From the user’s point of view, it acts like a local function call.  Step 1: A REST call is made from the client. The request body is usually in JSON format.  Steps 2 - 4: The order service (gRPC client) receives the REST call, transforms it, and makes an RPC call to the payment service. gPRC encodes the client stub into a binary format and sends it to the low-level transport layer.  Step 5: gRPC sends the packets over the network via HTTP2. Because of binary encoding and network optimizations, gRPC is said to be 5X faster than JSON.  Steps 6 - 8: The payment service (gRPC server) receives the packets from the network, decodes them, and invokes the server application.  Steps 9 - 11: The result is returned from the server application, and gets encoded and sent to the transport layer.  Steps 12 - 14: The order service receives the packets, decodes them, and sends the result to the client application. |

**Domain-Driven Design (DDD)**

|  |  |
| --- | --- |
| Diagram  Description automatically generated | DDD was introduced in Eric Evans’ classic book “Domain-Driven Design: Tackling Complexity in the Heart of Software”. It explained a methodology to model a complex business. There is a lot of content in this book, so I'll summarize the basics.  The composition of domain objects:  Entity: a domain object that has ID and life cycle.  Value Object: a domain object without ID. It is used to describe the property of Entity.  Aggregate: a collection of Entities that are bounded together by Aggregate Root (which is also an entity). It is the unit of storage.  The life cycle of domain objects:  Repository: storing and loading the Aggregate.  Factory: handling the creation of the Aggregate.  Behavior of domain objects:  Domain Service: orchestrate multiple Aggregate.  Domain Event: a description of what has happened to the Aggregate. The publication is made public so others can consume and reconstruct it. |

How do Apple Pay and Google Pay handle sensitive card info?

|  |  |
| --- | --- |
|  | To understand the difference, we break down the process into two flows.  1 Registering your credit card flow  2 Basic payment flow  1. The registration flow is represented by steps 1~3 for both cases. The difference is:  Apple Pay: Apple doesn’t store any card info. It passes the card info to the bank. Bank returns a token called DAN (device account number) to the iPhone. iPhone then stores DAN into a special hardware chip.  Google Pay: When you register the credit card with Google Pay, the card info is stored in the Google server. Google returns a payment token to the phone.  2. When you click the “Pay” button on your phone, the basic payment flow starts. Here are the differences:  Apple Pay: For iPhone, the e-commerce server passes the DAN to the bank.  Google Pay: In the Google Pay case, the e-commerce server passes the payment token to the Google server. Google server looks up the credit card info and passes it to the bank.  In the diagram, the red arrow means the credit card info is available on the public network, although it is encrypted. |

How do Google/Apple maps blur license plates and human faces on Street View

|  |  |
| --- | --- |
| No alt text provided for this image | The diagram below presents a possible solution that might work in an interview setting.  The high-level architecture is broken down into three stages:  Data pipeline - prepare the training data set  Blurring pipeline - extract and classify objects and blur relevant objects, for example, license plates and faces.  Serving pipeline - serve blurred street view images to users.  Data pipeline  Step 1: We get the annotated dataset for training. The objects are marked in bounding boxes.  Steps 2-4: The dataset goes through preprocessing and augmentation to be normalized and scaled.  Steps 5-6: The annotated dataset is then used to train the machine learning model, which is a 2-stage network.  Blurring pipeline  Steps 7-10: The street view images go through preprocessing, and object boundaries are detected in the images. Then sensitive objects are blurred, and the images are stored in an object store.  Serving pipeline  Step 11: The blurred images can now be retrieved by users. |

Why is Nginx called a “reverse” proxy Vs Forward Proxy?

|  |  |
| --- | --- |
| No alt text provided for this image | the differences between a forward proxy and a reverse proxy.  A forward proxy is a server that sits between user devices and the internet.  A forward proxy is commonly used for:   * Protect clients * Avoid browsing restrictions * Block access to certain content   A reverse proxy is a server that accepts a request from the client, forwards the request to web servers, and returns the results to the client as if the proxy server had processed the request.  A reverse proxy is good for:   * Protect servers * Load balancing * Cache static contents * Encrypt and decrypt SSL communications |

Why do we use column-based DB? Does column-based DB provide better performance?

|  |  |
| --- | --- |
| a close up of a chart | The diagram below shows how data is stored in column-based DB.  When to use   * The table is a wide table with many columns. * The queries and calculations are on a small number of columns. * A lot of the columns contain a few distinct values.   Benefits of column-based DB   * Higher data compression rates. * Higher performance on OLAP functions. * No need for additional indexes |

The Payments Ecosystem

|  |  |
| --- | --- |
|  | How do fintech startups find new opportunities among so many payment companies? What do PayPal, Stripe, and Square do exactly?  Steps 0-1: The cardholder opens an account in the issuing bank and gets the debit/credit card. The merchant registers with ISO (Independent Sales Organization) or MSP (Member Service Provider) for in-store sales. ISO/MSP partners with payment processors to open merchant accounts.  Steps 2-5: The acquiring process.  The payment gateway accepts the purchase transaction and collects payment information. It is then sent to a payment processor, which uses customer information to collect payments. The acquiring processor sends the transaction to the card network. It also owns and operates the merchant’s account during settlement, which doesn’t happen in real-time.  Steps 6-8: The issuing process.  The issuing processor talks to the card network on the issuing bank’s behalf. It validates and operates the customer’s account.  I’ve listed some companies in different verticals in the diagram. Notice payment companies usually start from one vertical, but later expand to multiple verticals. |

What is Web 3.0?

|  |  |
| --- | --- |
|  | The diagram below shows Web 1.0/Web 2.0/Web 3.0 from a bird's-eye view.  Web 1.0 - Read Only  Between 1991 and 2004, the internet is like a catalog of static pages. We can browse the content by jumping from one hyperlink to another. It doesn’t provide any interactions with the content.  Web 2.0 - Read Write  From 2004 to now, the internet has evolved to have search engines, social media apps, and recommendation algorithms backed apps.  Because the apps digitalize human behaviors and persist user data when users interact with these apps, big companies leverage user data for advertisements, which becomes one of the main business models in Web 2.0.  That’s why people say the apps know you better than your friends, family, or even yourself.  Web 3.0 - Read Write Own  The idea has been discussed a lot recently due to the development of blockchain and decentralized apps. The creators’ content is stored on IPFS (InterPlanetary File System) and owned by the users. If apps want to access the data, they need to get authorization from the users and pay for it.  In Web 3.0, the ownership change may lead to some major innovations. |

<https://redis.com/docs/8-data-modeling-patterns-in-redis/?utm_source=bytebytego&utm_medium=thirdparty&utm_campaign=growth-learning&utm_content=newsletter>

What does API gateway do?

|  |  |
| --- | --- |
| Diagram  Description automatically generated | The diagram below shows the detail.  Step 1 - The client sends an HTTP request to the API gateway.  Step 2 - The API gateway parses and validates the attributes in the HTTP request.  Step 3 - The API gateway performs allow-list/deny-list checks.  Step 4 - The API gateway talks to an identity provider for authentication and authorization.  Step 5 - The rate limiting rules are applied to the request. If it is over the limit, the request is rejected.  Steps 6 and 7 - Now that the request has passed basic checks, the API gateway finds the relevant service to route to by path matching.  Step 8 - The API gateway transforms the request into the appropriate protocol and sends it to backend microservices.  Steps 9-12: The API gateway can handle errors properly, and deals with faults if the error takes a longer time to recover (circuit break). It can also leverage ELK (Elastic-Logstash-Kibana) stack for logging and monitoring. We sometimes cache data in the API gateway. |

What is GraphQL? Is it a replacement for the REST API?

|  |  |
| --- | --- |
|  | The diagram below shows the quick comparison between REST and GraphQL.  GraphQL is a query language for APIs developed by Meta. It provides a complete description of the data in the API and gives clients the power to ask for exactly what they need.  GraphQL servers sit in between the client and the backend services.  GraphQL can aggregate multiple REST requests into one query. GraphQL server organizes the resources in a graph.  GraphQL supports queries, mutations (applying data modifications to resources), and subscriptions (receiving notifications on schema modifications). |

Kubernetes

A visual guide on troubleshooting Kubernetes deployments by learnk8s on Twitter.

Diagram, schematic

Description automatically generated

Live streaming explained

|  |  |
| --- | --- |
| Diagram  Description automatically generated | How do video live streamings work on YouTube, TikTok live, or Twitch? The technique is called live streaming.    Livestreaming differs from regular streaming because the video content is sent via the internet in real-time, usually with a latency of just a few seconds.    The diagram below explains what happens behind the scenes to make this possible.  Step 1: The raw video data is captured by a microphone and camera. The data is sent to the server side.    Step 2: The video data is compressed and encoded. For example, the compressing algorithm separates the background and other video elements. After compression, the video is encoded to standards such as H.264. The size of the video data is much smaller after this step.    Step 3: The encoded data is divided into smaller segments, usually seconds in length, so it takes much less time to download or stream.    Step 4: The segmented data is sent to the streaming server. The streaming server needs to support different devices and network conditions. This is called ‘Adaptive Bitrate Streaming.’ This means we need to produce multiple files at different bitrates in steps 2 and 3.    Step 5: The live streaming data is pushed to edge servers supported by CDN (Content Delivery Network.) Millions of viewers can watch the video from an edge server nearby. CDN significantly lowers data transmission latency.    Step 6: The viewers’ devices decode and decompress the video data and play the video in a video player.    Steps 7 and 8: If the video needs to be stored for replay, the encoded data is sent to a storage server, and viewers can request a replay from it later.    Standard protocols for live streaming include:   * RTMP (Real-Time Messaging Protocol): Originally developed by Macromedia to transmit data between a Flash player and a server. Now it is used for streaming video data over the internet. Note that video conferencing applications like Skype use RTC (Real-Time Communication) protocol for lower latency. * HLS (HTTP Live Streaming): It requires the H.264 or H.265 encoding. Apple devices accept only HLS format. * DASH (Dynamic Adaptive Streaming over HTTP): DASH does not support Apple devices. * Both HLS and DASH support adaptive bitrate streaming.   Some of the optimizations & Suitable type of storage for video persistence in step 7 |

<https://voyager.postman.com/infographics/2022-postman-api-platform-landscape-pm.png>

Diagram, schematic

Description automatically generated

Algorithms you should know before taking System Design Interviews

I put together a list and explained why they are important. Those algorithms are not only useful for interviews but good to understand for any software engineer.

Graphical user interface

Description automatically generated with medium confidence