

Kaigi on Rails 2025における subscreenの実装について

うなすけ

北陸Ruby会議01

2025-12-06



自己紹介



- Name: うなすけ
- Work: フリーランス
- Kaigi on Rails Organizer
- GitHub <https://github.com/unasuke>
- Fediverse <https://mstdn.unasuke.com/@unasuke>
- X https://x.com/yu_suke1994



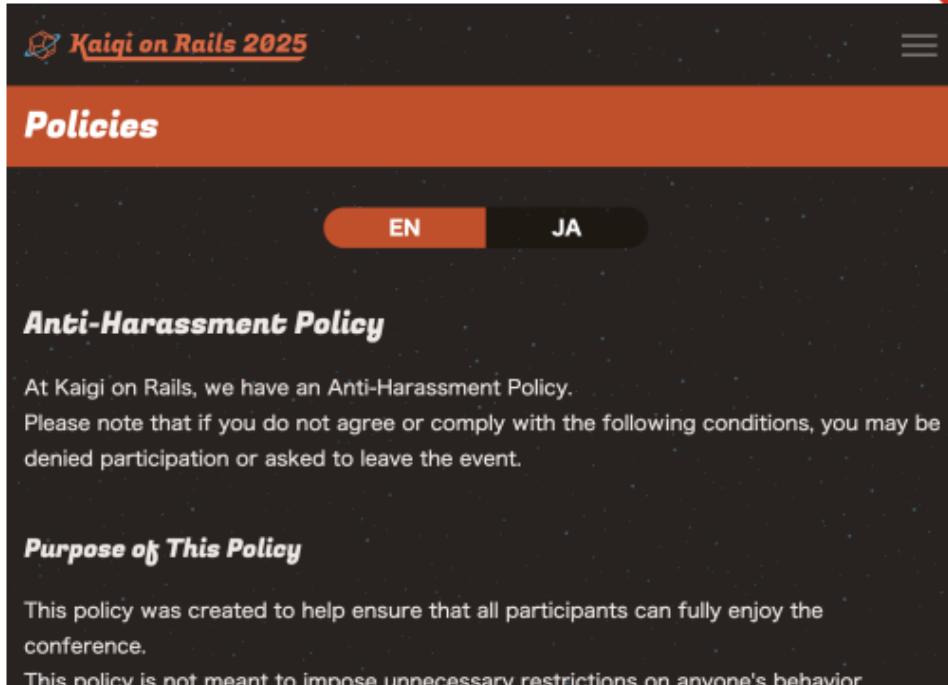
Kaigi on Railsとは



Kaigi on Rails is an approachable, inclusive, web-focused tech conference, designed to lower the barrier of entry to participating in conferences.

*Kaigi on Rails*のコアコンセプトは「初学者から上級者までが楽しめるWeb系の技術カンファレンス」です。

[cited from [`https://kaigionrails.org/2025/about/`](https://kaigionrails.org/2025/about/)]



Kaigi on Rails 2025

Policies

EN **JA**

Anti-Harassment Policy

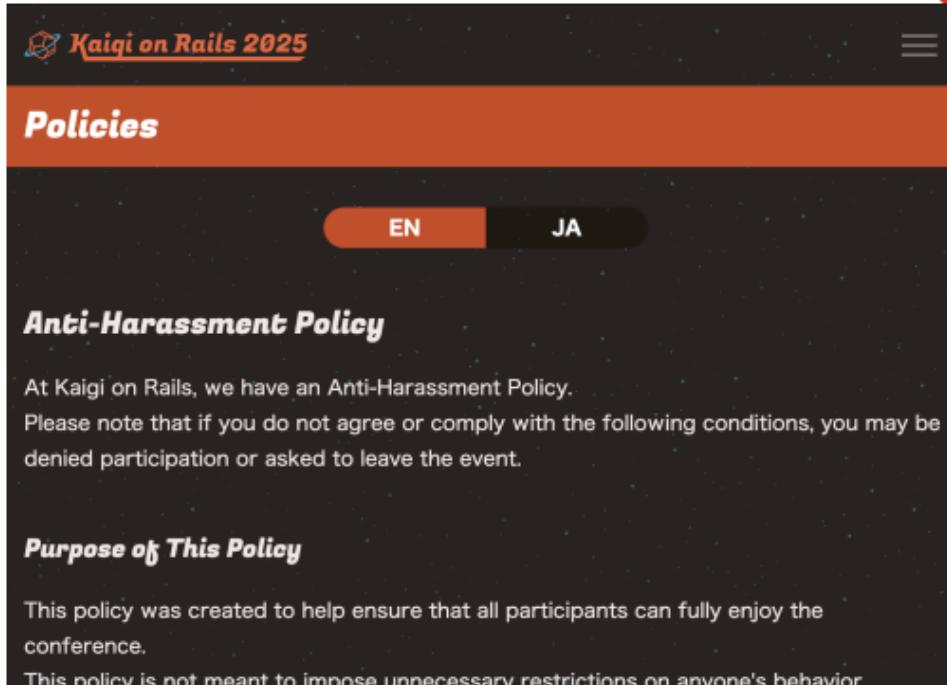
At Kaigi on Rails, we have an Anti-Harassment Policy. Please note that if you do not agree or comply with the following conditions, you may be denied participation or asked to leave the event.

Purpose of This Policy

This policy was created to help ensure that all participants can fully enjoy the conference.

This policy is not meant to impose unnecessary restrictions on anyone's behavior

EN/JA



Anti-Harassment Policy

At Kaigi on Rails, we have an Anti-Harassment Policy. Please note that if you do not agree or comply with the following conditions, you may be denied participation or asked to leave the event.

Purpose of This Policy

This policy was created to help ensure that all participants can fully enjoy the conference.

This policy is not meant to impose unnecessary restrictions on anyone's behavior.

国際カンファレンスっぽい！

Kaigi on Rails 2025は国際カンファレンス？



- 「これができるれば国際カンファレンス」という基準はない
 - と、思ってるけど……
- 海外からの参加者
- 英語による発表
- 英語によるアナウンス
- ビザ取得支援
- 同時通訳
- 字幕

Kaigi on Rails 2025は国際カンファレンス？



- 「これができるれば国際カンファレンス」という基準はない
 - と、思ってるけど……
- 海外からの参加者
- 英語による発表
- 英語によるアナウンス
- ビザ取得支援
- **同時通訳**
- **字幕**

- JA → EN の同時通訳
 - 発表数的に厳しい
- EN → JA の同時通訳
 - こちらを用意した
 - 全員分のレシーバーを用意するのは厳しい

同時通訳のリソースが十分ではない
→ 字幕も用意したいよね

カンファレンスと字幕



- VimConf 2024
 - <https://pocketalk.jp>
- Observability Conference Tokyo 2025
 - <https://www.cuckoo.so/jp/>

- RubyKaigi
 - <https://github.com/ruby-no-kai/takeout-app>
 - <https://github.com/ruby-no-kai/signage-app>
- 東京Ruby会議12
 - <https://github.com/tokyorubykaigi12/captioner>

Fork?

keynote speakers



Keynote Speakers



MOROHASHI
Kyosuke 



Samuel
Williams 

-  NEW POST
-  HOME
-  POSTS
-  TAGS
-  SETTINGS

#634 WIP 2025 / other / 技術的TODO [Edit](#)

SHOULD ~ MAY

- conference-appにi18nの仕組みをいれる
- [REDACTED]
- subscreen字幕
 - 折角なのでfiberとかfalconを活用したい、そのためにはconference-appの外に出してもいいと思っている
 - [#592: \[WIP\] 2025/other/同通\(subscreen\)設計雑メモ](#)
- [REDACTED]
 - [REDACTED]
- [REDACTED]

new Falcon app



- Falconのアプリって作ったことない
 - ありますか？KaigiEffectで触った人はいる？
- リアルタイム文字起こしのアプリって作ったことない
 - さっきのコードを読めばなんとなくわかるけど
- リアルタイムにデータを配信するアプリって……
 - Action Cableではない
 - WebSocketならちょっとあるけど、WebSocketなの？

□ Conference Real-Time Transcription System ▾

共有

Y

カンファレンスのサブスクリーンに、登壇者の話している内容の文字起こしと、その翻訳を表示する仕組みを開発したいです。文字起こしおよび翻訳は参加者のスマートフォンでも見れるようにするため、少なくとも1000以上のクライアントが接続します。設計及び実装についての相談にのっていただけますか？

カンファレンスでの大規模リアルタイム文字起こし・翻訳システムの設計について相談させていただきます。1000以上のクライアント接続を想定した場合、いくつかの重要な技術課題があります。

システム全体の設計アプローチ

マイクロサービス構成を推奨します：

1. 音声取得・文字起こしサービス

具体的にどの部分から実装を始めたいか、または特定の技術選択について詳しく相談したい部分はありますか？

Y

まずは文字起こしの内容をクライアントに配信するところについて設計を行いたいです。
Server-side eventの使用を検討しています。

Server-Sent Events (SSE) を使った文字起こし配信システムの設計について詳しく検討しましょう。1000+クライアント接続でのSSE配信は良い選択です。

この設計について、特に知りたい部分（パフォーマンス最適化、エラーハンドリング、セキュリティなど）はありますか？

Y

サーバー側実装は、言語はRubyを、サーバー実装としてはFalconを使います。

Ruby + Falconを使ったSSE配信システムの設計について詳しく説明します。FalconはAsync/Awaitベースの高性能なRubyサーバーで、大量の同時接続に適しています。

Falcon + SSE基本実装

Y

負荷試験はどのように行うべきでしょうか？

カンファレンス用SSEシステムの負荷試験について、段階的なアプローチで詳しく説明します。

負荷試験の段階的アプローチ

1. 基本性能試験 (100-500接続)

ruby



Y

abやk6を負荷試験に使うことについてはどう思いますか？

ab (Apache Bench) と k6について、SSEシステムの負荷試験における適用性を詳しく評価します。

abの評価

制限事項と課題

bash



ここまで議論について、ドキュメントにまとめてください。

SSE翻訳字幕配信システムの技術設計ドキュメントをまとめました。

SSE翻訳字幕配信システム設計書
ドキュメント

ドリーム翻訳字幕
配信システム設計書
目次
目次 1. プロジ
エクト概要
目次 2. シス
カンターンス
カントラーンス

SSE翻訳字幕配信システムの技術設計書をまとめました。この議論で扱った主要なポイントを包括的にカバーしています：

主な内容：

- システム全体のアーキテクチャ設計

Y 文字起こしについては、Amazon Transcribeを使用します。この情報をもとにドキュメントを更新してください。

Amazon Transcribeを使用した文字起こし実装に対応してドキュメントを更新します。

SSE翻訳字幕配信システム設計書

Amazon Transcribeを使用した文字起こし実装に対応してドキュメントを更新しました。

主な更新内容：

1. Amazon Transcribe統合設計の新セクション追加

Y

ここで設計してもらったアプリをclaude codeに作業してもらう場合にどのようなpromptを渡せばよいですか？

Claude Codeに効果的に作業してもらうための段階的なプロンプトを提案します。

初期セットアップ用プロンプト

SSE翻訳字幕配信システムを実装してください。CLAUDE.mdの仕様に従って、以下の順序で実装を進めてください：

1. オブジェクト指向プログラミング構造を作成してください。

Vibe Coding



Hokuriku
RubyKaigi

File Edit Selection View Go Run Terminal Help

File status [0h 0m]

transcribe_standalone.rb

transcribe_standalone U transcribe_client U transcribe_3 U socket U ring_transcribe_server U mock_transcribe_client U ring_transcribe_uni U

147 # git:chime:msg: update_channel_message(

148 chime_user: chime_user_id,

149 channel:channel_id,

150 message_id: message_id,

151 content: content,

152

153 else

154 p [used_channel.message, caption]

155 resp = git:chime:msg:used_channel_message(

156 chime_user: chime_user_id,

157 channel:channel_id,

158 content: content,

159 type: 'STANDARD',

160 persistence: 'PERSISTENT',

161

162 @id_map[caption.result_id] = resp.message_id

163 end

164

165 # @id_map.delete(caption.result_id) unless caption.is_partial

166

167 end

168

169 class StandaloneOutput < StreamOutput

170 def handle(caption)

171 | \$stderr.puts caption.to_h.to_json

172 end

173 end

174

175 track_slog = RbEnv[0]

176

177 chime_user_id = Conference.data.fetch(:chime).fetch(:app_user_id)

178 channel_id = Conference.data

179 .fetch(:tracks)

180 .fetch(:track_slog, {})

181 .fetch(:chime, {})

182 .fetch(:channel_id, nil)

183

184 RbEnv['HelloEnv'] chime_user_id: chime_user_id, channel_id: channel_id

185

186 # If track_slog nil channel_id

187 # raise "Define caption_channel_id"

188 end

189

190 input = \$fd.stdin.read

191 engine = TranscribeEngine.new

192 output = StandaloneOutput.new

193

194 input.on_data do |chunk|

195 | p on_audio: chunk.bytesize

196 engine.feed(chunk)

197 end

198

199 engine.on_transcript_event do |e|

200 | output.feed(e)

201 end

202

203 begin

204 | output.start

205 | call = engine.start

206 | input.start

207 | p call.wait.inspect

208

209

210 # 1000s graceful restart

211

212 begin

213 | output.start

214 | call = engine.start

215 | input.start

216 | p call.wait.inspect

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1461

1462

1463

1464

1465

1466

1467

1468

1469

1470

1471

1472

1473

1474

1475

1476

1477

1478

1479

1480

1481

1482

1483

1484

1485

1486

1487

1488

1489

1490

1491

1492

1493

1494

1495

1496

1497

1498

1499

1500

1501

1502

1503

1504

1505

1506

1507

1508

1509

1510

1511

1512

1513

1514

1515

1516

1517

1518

1519

1520

1521

1522

1523

1524

1525

1526

1527

1528

1529

1530

1531

1532

1533

1534

1535

1536

1537

1538

1539

1540

1541

1542

1543

1544

1545

1546

1547

Vibe Coding

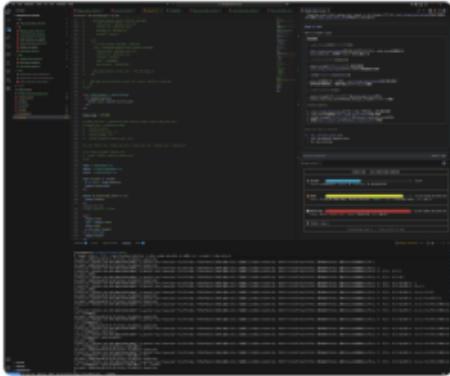


Vibe Coding (9/22)



9月22日 (月) ▾

unasuke/Organizer 15:01
おべべべべべ
image.png ▾



unasuke/Organizer 16:19
あばばばばば
image.png ▾



本番4日前にやることではない

Vibe codingまとめ



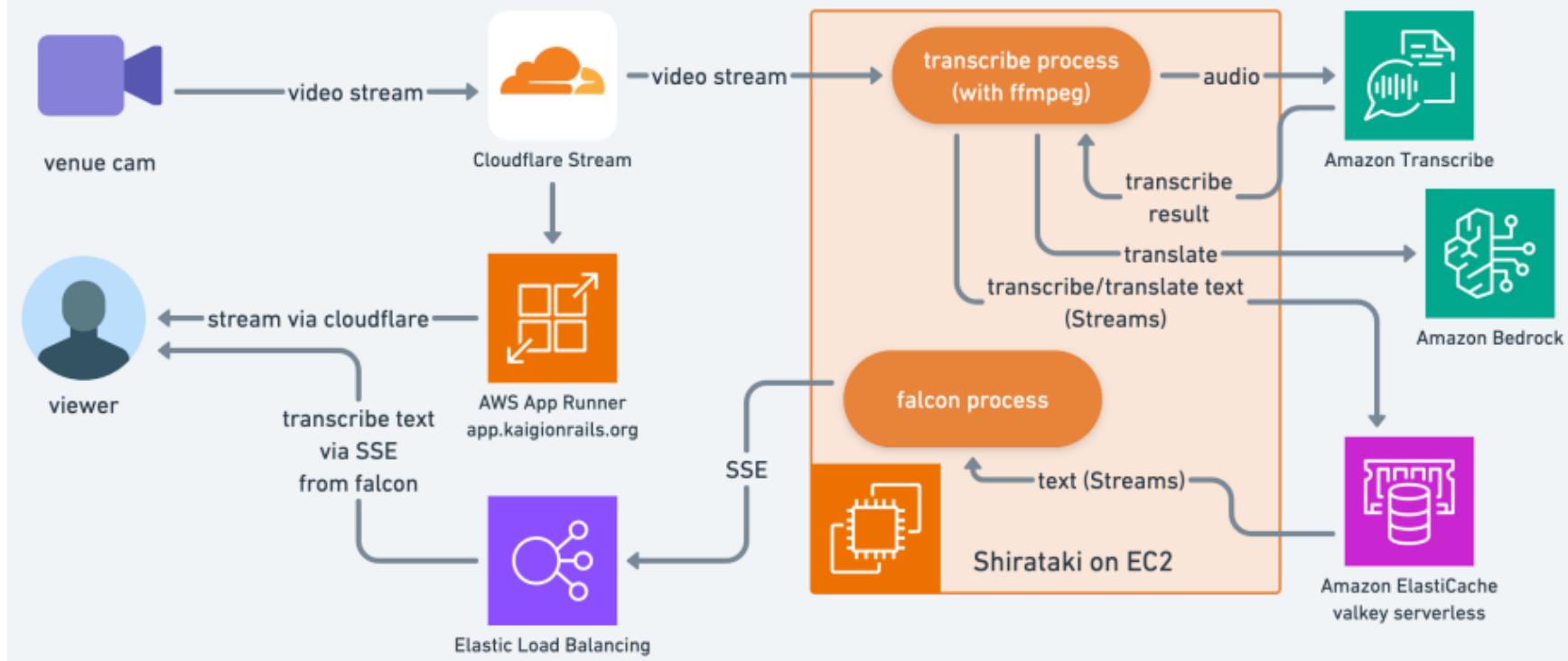
- まずはAIと壁打ちして方針決め
 - これは6月頃からやっていた
- 方針をもとに手元でVibe coding
 - Claude Max
- AWSの環境を構築するTerraformは人力
 - やるべきことがわかっているので

- <https://github.com/kaigionrails/shirataki>
 - ffmpegで配信を受け取り、音声をAmazon Transcribeに流す
 - Transcribeの結果をBedrockで英訳する
 - 日本語とAI訳をValkeyにstreamする
 - Valkey経由でstreamされた字幕情報をFalconがSSEする
- <https://github.com/kaigionrails/terraform/pull/18>
 - Valkey serverless
 - shiratakiが動くEC2
 - shiratakiにリクエストを流すALB
 - などなど

Kaigi on Rails 2025での構成



Hokuriku
RubyKaigi



トラブル集



- 本番環境で動かない
- 配信が止まると全てが止まる
- 同時通訳との兼ね合い
- (BedrockのAPI call quotaにひっかかる)

本番環境で動かないトラブル



- 検証環境では動いていたのに本番で動かない！
- ブラウザからのSSE受信ができない
- EC2上のプロセスは動いている

本番環境で動かないトラブル



- 検証環境では動いていたのに本番で動かない！
- ブラウザからのSSE受信ができない
- EC2上のプロセスは動いている
- 理由
 - 本番へのドメイン登録忘れ
 - ブラウザ(というかOS)がNXDOMAIN?をキャッシュ?
 - ドメイン登録とOS再起動で解決

配信が止まると全てが止まるトラブル



そらは

@sora_h



Blue部屋の幹線、会場借用してたけどFaultyと判断したので自前工事して
自前配線に切り替わりました [#rubykaigiNOC](#) [#kaigionrails](#)



そらは @sora_h · 9月26日

Blue部屋のパケロス率高いのは把握してるけど、被疑区間が自分たちの持ち物じ
ゃない有線なんだよな [#kaigionrails](#) [#rubykaigiNOC](#)

午後4:40 · 2025年9月26日 · 4,965 件の表示

https://x.com/sora_h/status/1971479957508587953

配信が止まると全てが止まるトラブル



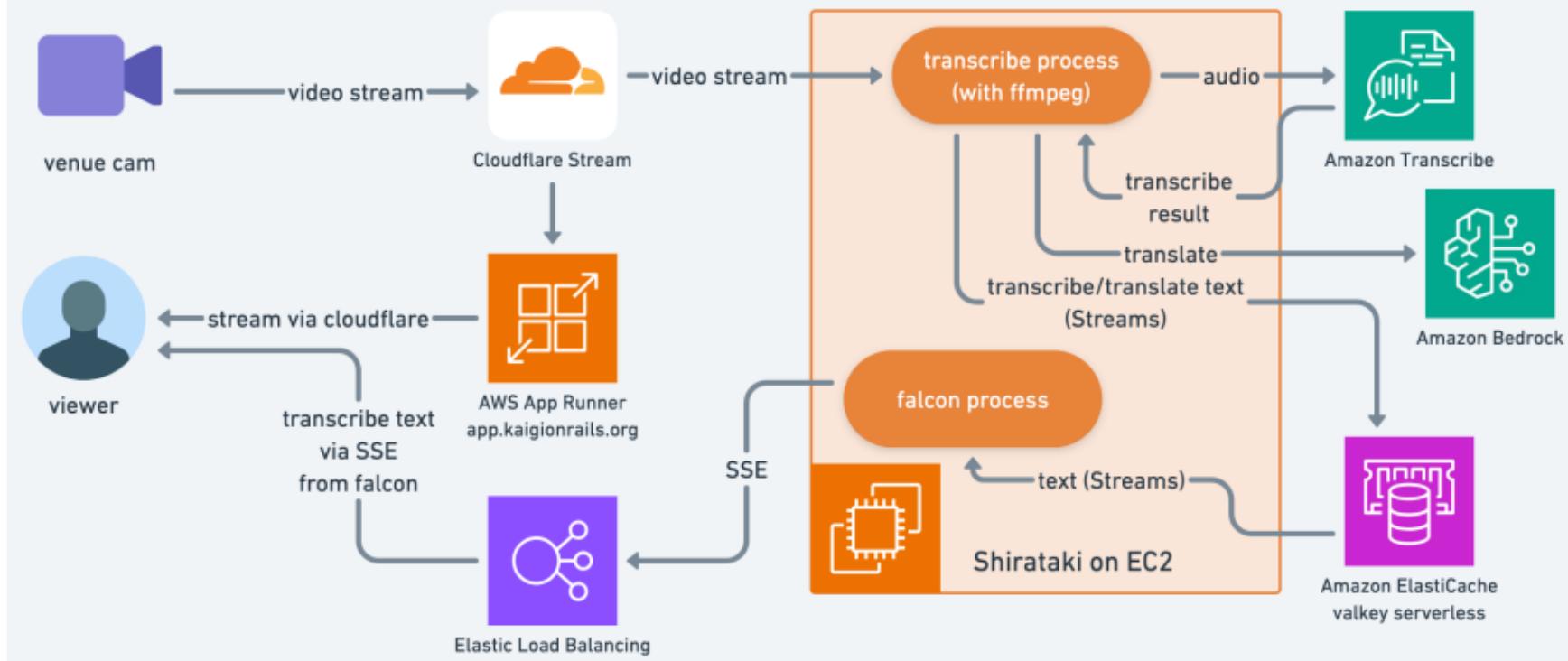
- インターネットへの接続が不安定
- Cloudflareへのストリーミングが止まる
- するとCloudflareからのストリーミングも止まる
- ffmpegによる受信プロセスが終了し、おしまい

配信が止まると全てが止まるトラブル



- インターネットへの接続が不安定
- Cloudflareへのストリーミングが止まる
- するとCloudflareからのストリーミングも止まる
- ffmpegによる受信プロセスが終了し、おしまい
- 解決策
 - 手元PCでログをずっと見守る
 - 止まつたら再起動
 - 部屋から「字幕止まってる」という報告があれば再起動

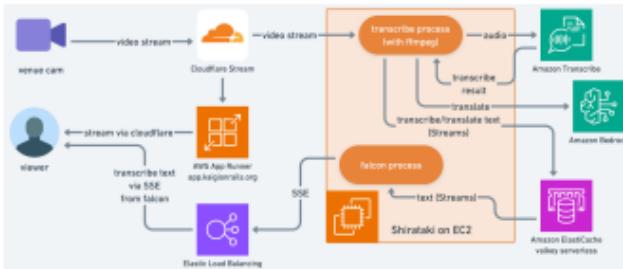
同時通訳セッションのトラブル



同時通訳セッションのトラブル



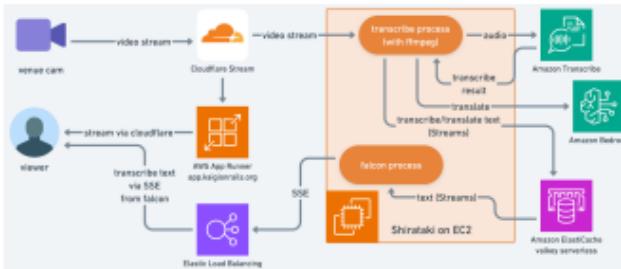
- 音声の言語は動的に変えられない
- 環境変数で現在の言語を指定している
- 同時通訳のセッションはJAもENも流れてくる
 - 翻訳がいらない



同時通訳セッションのトラブル



- 音声の言語は動的に変えられない
- 環境変数で現在の言語を指定している
- 同時通訳のセッションはJAもENも流れてくる
 - 翻訳がいらない
- 解決策
 - セッション切り替えのタイミングで素早く設定を切り替えて再起動
 - 英語セッションが終わったら戻す



なんとかKaigi on Rails 2025を乗り切った



- おおむね動いていた(と思う)
- 用意したリソースは十分だった
 - t3.mediumを4台だっけ？(うろおぼえ)
- ないよりマシ、くらいのものは提供できたのでは？

見えてきた課題



- 前述の問題
 - 配信が止まったときの復旧
 - 動的な翻訳の有無
- SSEの再接続
 - エラーになったら画面リロード
- メトリックを取る
 - 今何人見てるのか
- 屬人化
 - 自分の脳内にしかない手順

AIに書かせてみてどうだった？



- 時間がないときには本当に助かる
 - 質はまあ……それなり
- 自分のコードじゃない感じ
 - 責任は自分にあるのはそう
 - 「自分で書いた」感は本当に希薄

「自分で書きたい」という想い



- <https://github.com/unasuke/shirataki/tree/hokurikurk01>
 - 北陸Ruby会議01に向けて少し手を入れた
 - ffmpegの再起動
 - view
 - OpenTelemetry
 - このdiffはほぼ手で書いた



たのしい

Ruby

第6版

高橋征義・後藤裕蔵——著

Masayoshi Takahashi, Yuuzou Gotou

まつもとゆきひろ——監修

Yukihiro Matsumoto

まとめっぽく

- 自分たちのカンファレンスは自分たちで作りたい
- カンファレンスをやるのは楽しい
- Rubyを書くのは楽しい

