ARCHITECTURE COMPARISON (v2.0 : v3.0)

Model Mapping & Migration Strategy

Sandbox 2.0 Model	Sandbox 3.0 Model	Change Type	Migration Action
User	User	✅ Keep	No changes needed
Account	Account	✅ Keep	OAuth support unchanged
Session	Session	✅ Keep	Session management same
ActivateToken	ActivateToken	✅ Keep	Email verification same
ResetToken	ResetToken	✅ Keep	Password reset same
TicketCompetition	CompetitionRegistration	🔄 Rename + Modify	Rename table, add UNIQUE(userId), change field names
Team	Team	🔄 Modify	Remove teamCode field, keep structure

ParticipantCompetition	TeamMember	🔄 Rename + Modify	Rename table, add proofOfRegistrationLink, add UNIQUE(email)
Abstract	PreliminarySubmission	🔄 Rename + Modify	Rename table, generalize for all competitions
Regist3Data	Payment	🔄 Split	Extract payment-only fields, remove final work fields
Karya	❌ Remove	No voting system in v3.0	
PTCSubmissions	SemifinalSubmission	🔄 Merge	Unified model for all competitions with competition-specific fields
H4HSubmissions	SemifinalSubmission	🔄 Merge	Same as above
❌ Not in v2.0	Competition	➕ New	Config table for competition metadata and deadlines

❌ Not in v2.0	FinalSubmission	➕ New	BCC final phase submission
Admin	Admin	✅ Keep	Reuse with same role structure
TicketExhibition	❌ Remove	Different event (Main Event tickets)	
TicketGS	❌ Remove	Different event	
RegisExhiData	❌ Remove	Different event	
TransactionDetail	❌ Remove	No Midtrans integration in v3.0	
GrandSeminar	❌ Remove	Different event	
RefferalCode	❌ Remove	Not in requirements	

Key Code Adjustment Points

1. API Routes Changes

Remove (v2.0 only):

```text
❌ /api/ticket/exhibition
❌ /api/ticket/gs
❌ /api/voting
❌ /api/refferal-code
```

Rename/Modify:

```text
🔄 /api/ticket/competition → /api/team/register
🔄 /api/regist2 → /api/submission/preliminary
🔄 /api/regist3 → Split into:
  - /api/payment/submit
  - /api/submission/semifinal
```

Create New:

```text
✅ /api/competition/[code] - Fetch competition config
✅ /api/submission/final - BCC final submission
✅ /api/admin/registration/[id]/verify - Approve/reject registration
✅ /api/admin/submission/preliminary/[id]/review - Review abstract
✅ /api/admin/payment/[id]/verify - Verify payment
✅ /api/admin/finalists/select - Bulk select finalists
```

Keep (No Changes):

```text
✅ /api/auth/* - All auth routes
✅ /api/uploads - File upload route
✅ /api/team/[teamId] - Fetch team data (adjust fields)
```

---

2. Database Schema Changes

Critical Migrations:

```sql
-- 1. Add unique constraint: one user = one registration
ALTER TABLE CompetitionRegistration
ADD CONSTRAINT unique_user_registration UNIQUE (userId);
```

```sql
-- 2. Add unique constraint: one email = one team membership
ALTER TABLE TeamMember
ADD CONSTRAINT unique_member_email UNIQUE (email);

-- 3. Rename TicketCompetition to CompetitionRegistration
ALTER TABLE TicketCompetition
RENAME TO CompetitionRegistration;

-- 4. Add new fields to CompetitionRegistration
ALTER TABLE CompetitionRegistration
ADD COLUMN verificationStatus VARCHAR(20) DEFAULT 'pending',
ADD COLUMN currentPhase VARCHAR(20) DEFAULT 'registration',
ADD COLUMN isPreliminaryQualified BOOLEAN DEFAULT false,
ADD COLUMN isSemifinalQualified BOOLEAN DEFAULT false;

-- 5. Rename ParticipantCompetition to TeamMember
ALTER TABLE ParticipantCompetition
RENAME TO TeamMember;

-- 6. Add proofOfRegistrationLink to TeamMember
ALTER TABLE TeamMember
ADD COLUMN proofOfRegistrationLink VARCHAR(500) NOT NULL;

-- 7. Create Competition config table
CREATE TABLE Competition (
  id VARCHAR(50) PRIMARY KEY,
  code VARCHAR(10) UNIQUE NOT NULL,
  name VARCHAR(100) NOT NULL,
  minTeamSize INT NOT NULL,
  maxTeamSize INT NOT NULL,
  registrationFee DECIMAL(10,2) DEFAULT 0,
  preliminaryDeadline DATE NOT NULL,
  paymentDeadline DATE NOT NULL,
  semifinalDeadline DATE NOT NULL,
  finalDeadline DATE,
  isActive BOOLEAN DEFAULT true
);

-- 8. Seed Competition data
INSERT INTO Competition VALUES
('comp_ptc', 'PTC', 'ProtoTech Contest', 3, 5, 150000, '2026-02-15', '2026-03-01',
'2026-03-30', NULL, true),
('comp_tpc', 'TPC', 'Technovate Paper Competition', 1, 3, 100000, '2026-02-15',
'2026-03-01', '2026-03-30', NULL, true),
('comp_bcc', 'BCC', 'Business Case Competition', 3, 3, 100000, '2026-02-15',
'2026-03-01', '2026-03-30', '2026-04-15', true);

-- 9. Create SemifinalSubmission (merge PTC & H4H submissions)
CREATE TABLE SemifinalSubmission (
  id VARCHAR(50) PRIMARY KEY,
  registrationId VARCHAR(50) UNIQUE NOT NULL,
  competitionType VARCHAR(10) NOT NULL,
  fullPaperUrl VARCHAR(500),
  videoPitchingUrl VARCHAR(500),
  fullProposalUrl VARCHAR(500),
  fullCaseProposalUrl VARCHAR(500),
  submittedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- 10. Create FinalSubmission (BCC only)
CREATE TABLE FinalSubmission (
  id VARCHAR(50) PRIMARY KEY,
  registrationId VARCHAR(50) UNIQUE NOT NULL,
  pitchDeckUrl VARCHAR(500) NOT NULL,
  fileName VARCHAR(255) NOT NULL,
  fileSize BIGINT,
  submittedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

---

3. Session Callback Changes

v2.0 Session Structure:

typescript

```typescript
session.user = {
  id: string,
  username: string,
  email: string,
  ticketsCompetition: [ // ARRAY of tickets
    {
      id: string,
      competitionType: 'PTC' | 'H4H',
      verified: 'pending' | 'verified',
      stage: 1.0 | 2.0 | 3.0,
      teamId: string
    }
  ]
}
```

v3.0 Session Structure:

typescript
```
session.user = {
  id: string,
  username: string,
  email: string,
  registration: { // SINGLE registration object (not array)
    id: string,
    competitionType: 'PTC' | 'TPC' | 'BCC',
    verificationStatus: 'pending' | 'approved' | 'rejected',
    currentPhase: 'registration' | 'preliminary' | 'semifinal' | 'final',
    isPreliminaryQualified: boolean,
    isSemifinalQualified: boolean
  },
  team: {
    id: string,
    teamName: string,
    institution: string
  }
}
```

Migration Code:

typescript
```
// src/lib/authOptions.ts - Session callback

// v2.0 (OLD)
const tickets = await prisma.ticketCompetition.findMany({
  where: { userId: token.sub }
});
session.user.ticketsCompetition = tickets;

// v3.0 (NEW)
const registration = await prisma.competitionRegistration.findUnique({
  where: { userId: token.sub },
  include: {
    competition: true,
    team: true
  }
});

if (registration) {
  session.user.registration = {
    id: registration.id,
    competitionType: registration.competition.code,
    verificationStatus: registration.verificationStatus,
    currentPhase: registration.currentPhase,
    isPreliminaryQualified: registration.isPreliminaryQualified,
    isSemifinalQualified: registration.isSemifinalQualified
  };
  session.user.team = {
    id: registration.team.id,
    teamName: registration.team.teamName,
    institution: registration.team.institution
  };
}
```

---

4. Google Sheets Integration Changes

v2.0: Single webhook untuk semua data

typescript

```typescript
const SHEET_WEBHOOK = process.env.GOOGLE_SHEET_WEBHOOK_URL;
```

v3.0: Multiple webhooks per submission type

typescript

```typescript
const SHEET_WEBHOOKS = {
  registration: process.env.SHEET_WEBHOOK_REGISTRATION,
  preliminary: process.env.SHEET_WEBHOOK_PRELIMINARY,
  payment: process.env.SHEET_WEBHOOK_PAYMENT,
  semifinal: process.env.SHEET_WEBHOOK_SEMIFINAL,
  final: process.env.SHEET_WEBHOOK_FINAL
};

// Usage
await fetch(SHEET_WEBHOOKS.registration, {
  method: 'POST',
  body: JSON.stringify({
    teamName,
    institution,
    members: [...],
    competitionType,
    timestamp: new Date().toISOString()
  })
});
```

Benefits:

- Separate sheets for better organization
- Easier data analysis per stage
- Independent webhook failure (1 fails, others still work)

---

5. File Upload Strategy

v2.0: All files via UploadThing

typescript

```
// Everything is file upload
abstractFile: File → UploadThing → URL
videoFile: File → UploadThing → URL (besar!)
paymentProof: File → UploadThing → URL
```

v3.0: Hybrid approach

typescript

```
// Files
abstractFile: File → UploadThing → URL
fullPaper: File → UploadThing → URL
proposal: File → UploadThing → URL
paymentProof: File → UploadThing → URL
pitchDeck: File → UploadThing → URL

// YouTube link (no upload)
videoPitchingUrl: string (YouTube link only)
```

Reason: Video files sangat besar (100MB+), lebih efisien pakai YouTube. Tim upload video ke YouTube dulu, lalu paste link.