



Architecture Document - MedMeet Sprint 2

System Architecture – MedMeet VR

תאריך מסמך ארכיטקטורה 21: בינואר 2026
גרסה 2.0:

מנחה: ליבר ויצמן (Architect)

משתתפים:

- Architect: ליבר ויצמן
- Scrum Master: דוד רן בהן
- Product Owner: לידור בן סימון
- Quality Owner: רותם סויסה

תוכן עניינים

1. [System Overview](#)
2. [High-Level Architecture](#)
3. [Component Breakdown - Sprint 2](#)
4. [Technology Stack](#)
5. [Network Architecture](#)
6. [Data Flow Diagrams](#)
7. [Multi-Room System](#)
8. [AI Integration Architecture](#)
9. [Recording System](#)
10. [Performance Architecture](#)
11. [Security Considerations](#)
12. [Scalability](#)
13. [Technical Debt & Future](#)



Architecture Document - MedMeet Sprint 2

System Overview .1

1.1 מהו MedMeet VR?

MedMeet VR היא פלטפורמה למפגשים רפואיים במציאות מדומה המאפשרת לצוותים רפואיים לשתף פעולה מכל מקום בעולם.

1.2 Key Features (Sprint 2)

Core Infrastructure (Sprint 1):

- VR meeting room with avatars
- Real-time voice communication
- 3D medical model display
- Automatic session logging
- Sidebar menu with controls

Advanced Features (Sprint 2):

- Multi-Room System with navigation
- Full 3D model interaction (rotate, zoom, select)
- Collaboration tools (Laser, Annotations, Screenshots)
- Medical image display (PNG, JPG, X-ray)
- AI transcription (Whisper API)
- People Analytics
- 3D Recording & Playback
- Avatar improvements (selection, ground placement)

1.3 Platform Support

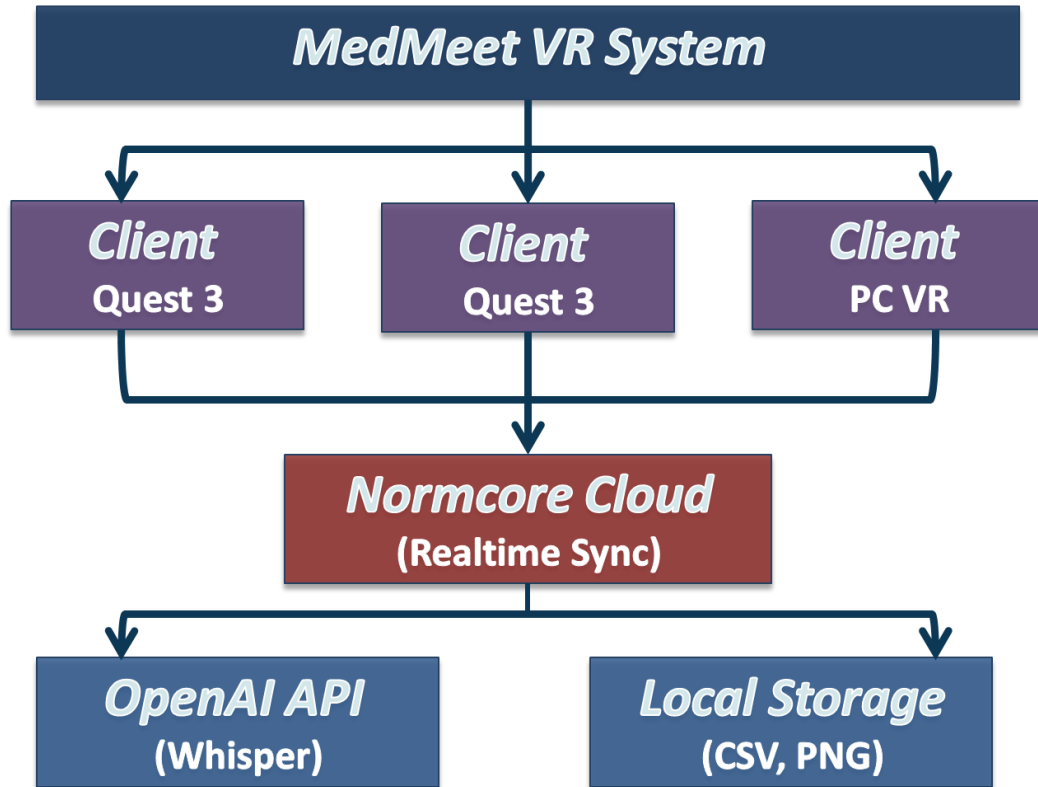
- **Development:** Unity 6.2 (6000.2.13f1)
- **Primary Target:** Meta Quest 3
- **Secondary Target:** PC VR (SteamVR, Oculus Runtime)
- **Networking:** Normcore 2.17.3
- **Cloud Services:** OpenAI API (Whisper)



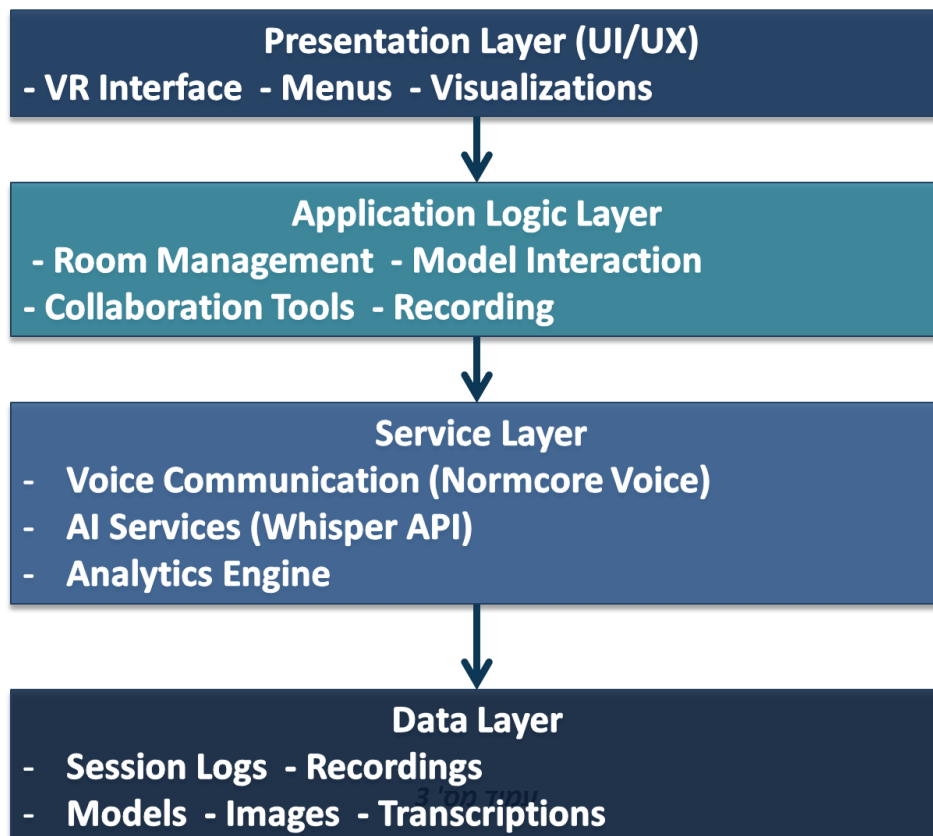
Architecture Document - MedMeet Sprint 2

High-Level Architecture .2

2.1 System Architecture Diagram



2.2 Architecture Layers





Architecture Document - MedMeet Sprint 2

Component Breakdown - Sprint 2 .3

3.1 Multi-Room System

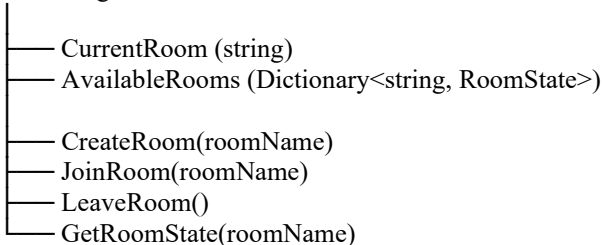
Component: RoomManager.cs

Responsibilities:

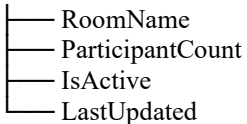
- Managing multiple VR rooms
- Room creation and joining
- Room state tracking
- Navigation between rooms

Architecture:

RoomManager



RoomState



Key Technologies:

- Normcore Room Management
- Unity Scene Management
- PlayerPrefs for room persistence

Integration Points:

- LoginManager → RoomManager (room selection)
- DoorController → RoomManager (navigation)
- Normcore Realtime → Room sync



Architecture Document - MedMeet Sprint 2

3.2 Advanced 3D Interaction System

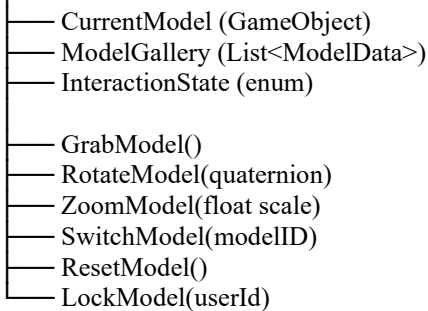
Component: ModelInteractionManager.cs

Responsibilities:

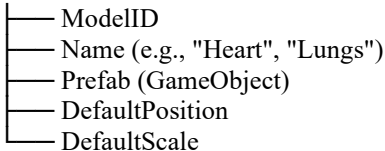
- 360° model rotation
- Zoom in/out
- Model selection from gallery
- Lock mechanism (single controller)
- Reset to default position

Architecture:

ModelInteractionManager

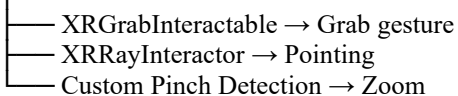


ModelData



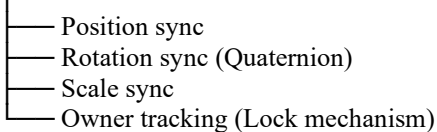
Gesture Detection:

XR Interaction Toolkit



Synchronization:

Normcore RealtimeTransform



עמוד מס' 5



Architecture Document - MedMeet Sprint 2

3.3 Collaboration Tools Suite

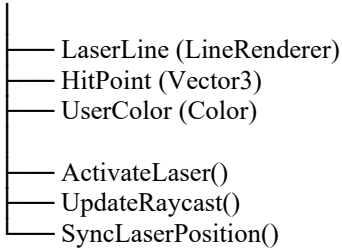
Components:

- LaserPointerController.cs
- AnnotationManager.cs
- ScreenshotCapture.cs

3.3.1 Laser Pointer

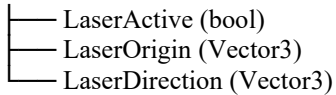
Architecture:

LaserPointerController



Normcore Sync:

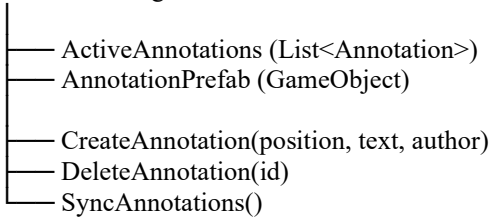
RealtimeView



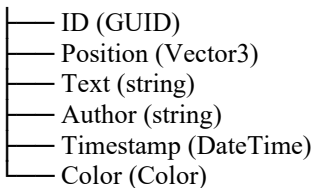
3.3.2 Annotation System

Architecture:

AnnotationManager



Annotation



Mesh Marking:

Paint-on-Mesh System

עמוד מס' 6



Architecture Document - MedMeet Sprint 2

- RaycastHit → Surface point
- Vertex Painter → Color vertices
- Normcore → Sync painted vertices

3.3.3 Screenshot System

Architecture:

ScreenshotCapture

- CaptureCamera (Camera)
- Resolution (1920x1080)
- CaptureScreen()
- SaveToPNG(texture, filename)
- AddToGallery(filepath)

Filename Format:

MedMeet_YYYY-MM-DD_HH-MM-SS.png

Example: MedMeet_2026-01-21_14-35-22.png

3.4 AI Integration Layer

Component: AITranscriptionService.cs

Architecture:

AITranscriptionService

- WhisperAPIClient
- AudioRecorder
- TranscriptionQueue
- StartRecording()
- StopRecording()
- SendToAPI(audioFile)
- ProcessResponse(json)
- ExportTranscript(format)

Data Flow:

Microphone → AudioClip
 → WAV Encoding
 → HTTP POST to Whisper API
 → JSON Response
 → Text Display + Storage

API Integration:

// Simplified Architecture

```
public class WhisperAPIClient
```

```
{
```

```
    private const string API_URL = "https://api.openai.com/v1/audio/transcriptions";
```

```
    public async Task<string> TranscribeAsync(byte[] audioData)
```

עמוד מס' 7



Architecture Document - MedMeet Sprint 2

```
{
  // 1. Create multipart form data
  var formData = new MultipartFormDataContent();
  formData.Add(new ByteArrayContent(audioData), "file", "audio.wav");
  formData.Add(new StringContent("whisper-1"), "model");
  formData.Add(new StringContent("he"), "language"); // Hebrew

  // 2. Send request
  var response = await httpClient.PostAsync(API_URL, formData);

  // 3. Parse response
  var json = await response.Content.ReadAsStringAsync();
  var result = JsonUtility.FromJson<WhisperResponse>(json);

  return result.text;
}
```

Error Handling:

Try-Catch Wrapper

- Network Error → Display: "Check internet connection"
- API Error (4xx) → Display: "Service unavailable"
- Timeout → Retry logic (max 3 attempts)
- Success → Display transcription

3.5 People Analytics Engine

Component: PeopleAnalyticsManager.cs

Architecture:

PeopleAnalyticsManager

- ParticipantData (Dictionary<userId, Analytics>)
- SessionStartTime (DateTime)
- TrackVoiceActivity(userId, isSpeaking)
- CalculateTalkTime(userId)
- CalculateParticipation(userId)
- GenerateReport()
- ExportToCSV()

ParticipantAnalytics

- UserID
- Name
- TotalTalkTime (TimeSpan)
- ParticipationPercentage (float)
- InteractionCount (int)
- LastActive (DateTime)



Architecture Document - MedMeet Sprint 2

Voice Detection:

Normcore Voice

- OnSpeakingStarted → StartTimer(userId)
- OnSpeakingStopped → StopTimer(userId)
- OnVolumeChanged → Track activity level

Analytics Calculation:

Participation % = (User Talk Time / Total Meeting Time) × 100

CSV Export Format:

Session ID, User Name, Role, Talk Time (sec), Participation %, Join Time, Leave Time

session_123, Dr. Cohen, Surgeon, 450, 45.0%, 14:30:00, 15:30:00

session_123, Dr. Weizman, Cardiologist, 300, 30.0%, 14:30:00, 15:30:00

3.6 Advanced Recording System

Component: RecordingManager.cs

Architecture:

RecordingManager

- IsRecording (bool)
- CurrentRecording (RecordingData)
- RecordingsList (List<RecordingData>)
- StartRecording()
- StopRecording()
- SaveRecording(version)
- LoadRecording(id)
- PlaybackRecording(id)

RecordingData

- RecordingID (GUID)
- Version (int)
- Timestamp (DateTime)
- Duration (TimeSpan)
- ParticipantCount (int)
- AvatarTransforms (List<TransformData>[])
- AudioData (byte[])
- InteractionEvents (List<Event>)
- FilePath (string)

What Gets Recorded:

Recording Components:

- Avatar Positions (Vector3[])
- Avatar Rotations (Quaternion[])
- Hand Positions (Vector3[])
- Audio Stream (WAV)
- Model Interactions (Events)

עמוד מס' 9



Architecture Document - MedMeet Sprint 2

- Annotations Created (Events)
- Laser Pointer Activity (Events)

Frame Rate: 30 FPS (optimized for file size)

Playback Architecture:

PlaybackController

- RecordingTimeline (float currentTime)
- PlaybackSpeed (float)
- LoadRecording(data)
- Play()
- Pause()
- Seek(time)
- SpawnGhostAvatars()

Ghost Avatars:

- Non-interactive replicas
- Follow recorded transforms
- Semi-transparent appearance

3.7 Avatar System v2

Component: AvatarManager.cs

Improvements:

- ✓ Gender selection (Male/Female)
- ✓ Ground placement fix (no floating)
- ✓ Basic hand tracking
- ✓ Walk/Idle animations

Architecture:

AvatarManager

- AvatarPrefabs (Dictionary<AvatarType, GameObject>)
- CurrentAvatar (GameObject)
- SelectAvatar(type)
- SpawnAvatar(position)
- FixGroundPlacement()
- SyncAvatarState()

AvatarType (enum)

- MaleDoctor
- FemaleDoctor
- MaleSurgeon
- FemaleSurgeon

Ground Placement Fix:

עמוד מס' 10



Architecture Document - MedMeet Sprint 2

```
// Before (Sprint 1): Floating
transform.position = new Vector3(x, y, z);

// After (Sprint 2): Grounded
void FixGroundPlacement()
{
    RaycastHit hit;
    if (Physics.Raycast(transform.position, Vector3.down, out hit, 10f))
    {
        transform.position = hit.point; // On floor
    }
}
```

Hand Tracking:

XR Hand Controller

- Left Hand Transform → Left Avatar Hand
- Right Hand Transform → Right Avatar Hand
- IK System → Natural movement

Animation Blend Tree:

Idle (50%) ↔ Walking (50%)
Based on: Velocity magnitude



Architecture Document - MedMeet Sprint 2

4. Technology Stack

4.1 Frontend - Unity VR

Unity 6.2 Engine
Rendering: URP (Universal)
VR Framework: XR Interaction
Physics: Unity Physics
Audio: Unity Audio + Normcore
UI: TextMeshPro + Canvas

Key Packages:

- com.unity.xr.interaction.toolkit - VR interactions
- com.unity.render-pipelines.universal - Graphics
- com.unity.textmeshpro - UI text
- com.unity.xr.management - XR settings

4.2 Networking - Multiplayer

Normcore 2.17.3
• Realtime (Scene Sync) • RealtimeView (Component Sync) • RealtimeTransform (Transforms) • RealtimeAvatar (Avatars) • Normcore Voice (Audio)

Cloud Infrastructure:

- Auto Region Selection
- WebRTC for voice
- 5 concurrent users max

4.3 AI Services

OpenAI Whisper API
• Model: whisper-1 • Languages: Hebrew, English • Format: WAV, MP3 • Max Size: 25MB

Request Format:

POST /v1/audio/transcriptions

Headers:

Authorization: Bearer sk-...

Body:

file: audio.wav

model: whisper-1



Architecture Document - MedMeet Sprint 2

language: he

4.4 Data Storage

Local Storage (Quest/PC):

- Session Logs → CSV files
- Recordings → Custom binary format
- Screenshots → PNG (1920x1080)
- Transcriptions → TXT
- Analytics → CSV

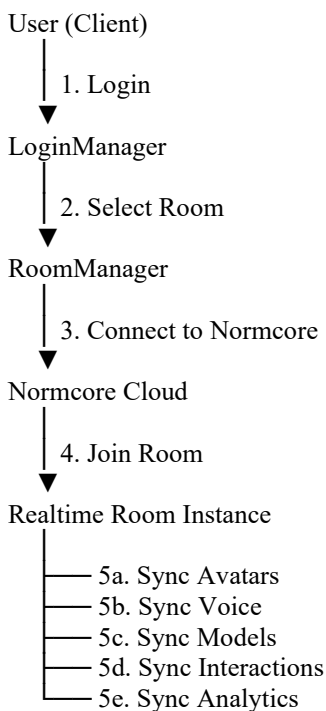
File Structure:

```

/MedMeet/
  /Logs/
    Session_2026-01-21.csv
  /Recordings/
    Recording_v1_2026-01-21_14-30.mrec
  /Screenshots/
    MedMeet_2026-01-21_14-35-22.png
  /Transcriptions/
    Transcript_2026-01-21_14-30.txt
  
```

5. Network Architecture

5.1 Connection Flow



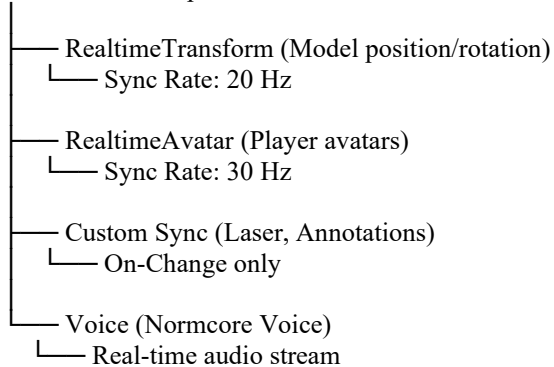


Architecture Document - MedMeet Sprint 2

5.2 Data Sync Architecture

Normcore Synchronization:

RealtimeView Components:



Bandwidth Usage:

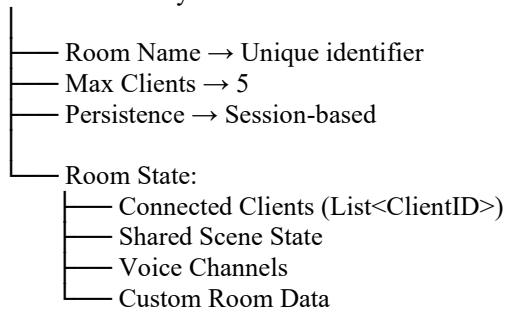
Per User:

- Voice: ~50 Kbps (compressed)
- Transforms: ~10 Kbps
- Custom Data: ~5 Kbps
- Total: ~65 Kbps per user

5 Users: ~325 Kbps total

5.3 Room Management

Normcore Room System:

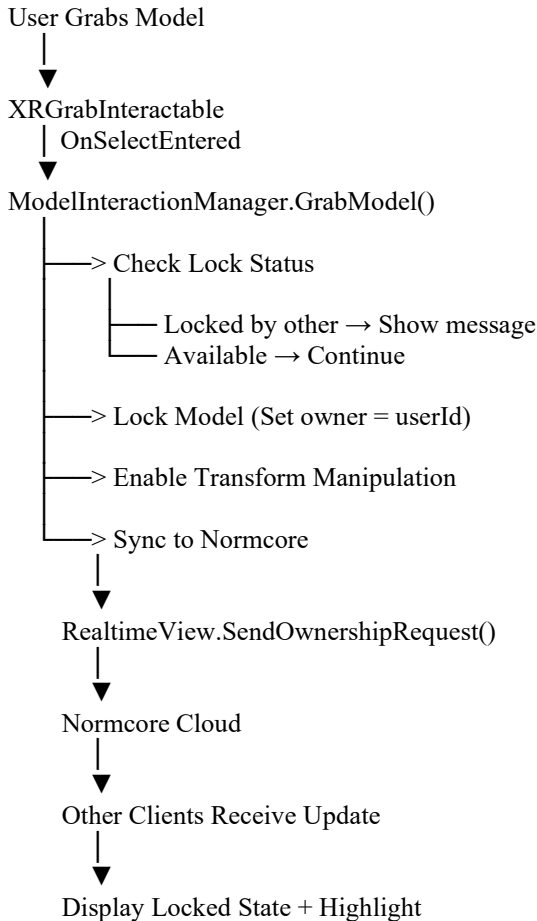




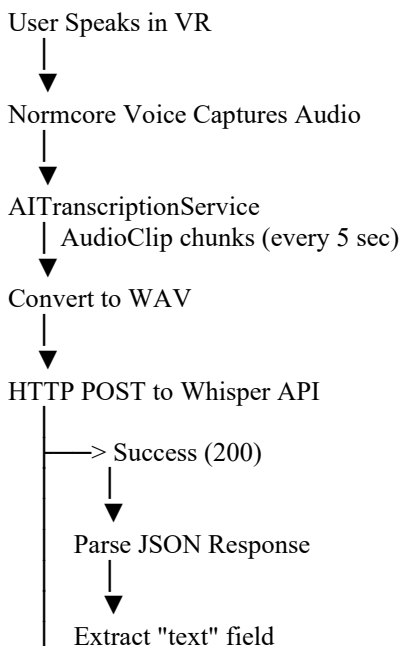
Architecture Document - MedMeet Sprint 2

6. Data Flow Diagrams

6.1 Model Interaction Flow

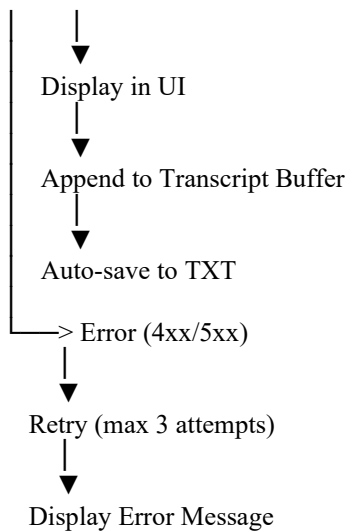


6.2 AI Transcription Flow

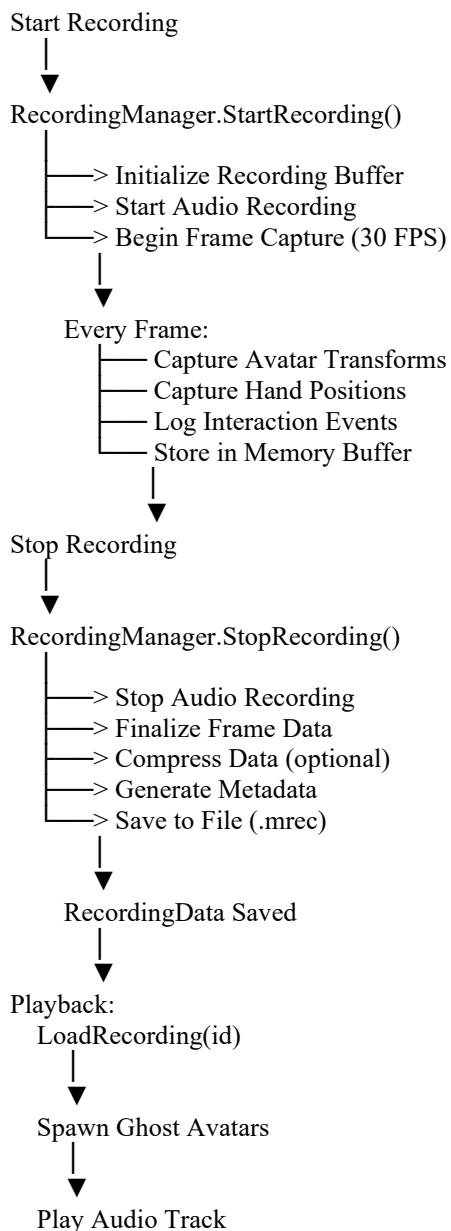




Architecture Document - MedMeet Sprint 2



6.3 Recording & Playback Flow





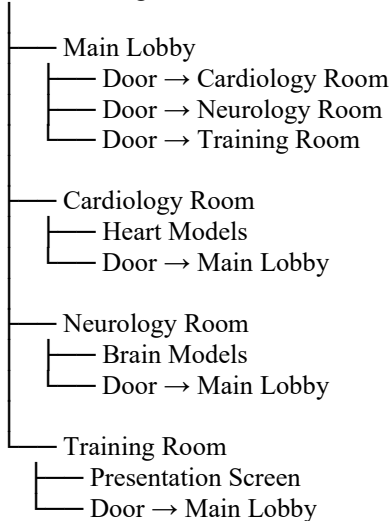
Architecture Document - MedMeet Sprint 2

Animate Transforms Frame-by-Frame

7. Multi-Room System

7.1 Room Architecture

Virtual Building:



7.2 Room Switching Implementation

```

public class DoorController : MonoBehaviour
{
    [SerializeField] private string targetRoomName;
    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            // Show UI prompt
            UIManager.ShowPrompt($"Enter {targetRoomName}?");
        }
    }
    public void OnDoorActivated()
    {
        // Disconnect from current room
        RoomManager.Instance.LeaveRoom();
        // Connect to new room
        RoomManager.Instance.JoinRoom(targetRoomName);
        // Load new scene (optional)
        SceneManager.LoadSceneAsync(targetRoomName);
    }
}
  
```

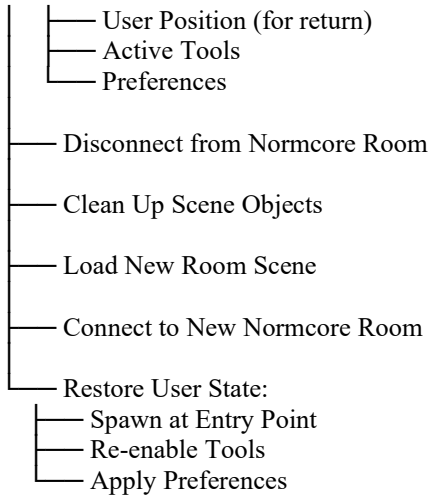
7.3 State Preservation

When User Switches Rooms:





Architecture Document - MedMeet Sprint 2



8. AI Integration Architecture

8.1 Whisper API Integration

Request Architecture:

Unity Client

HTTP Request

```

UnityWebRequest
Method: POST
URL: api.openai.com
Headers:
- Authorization: Bearer
- Content-Type: multipart
Body:
- file: audio.wav
- model: whisper-1
- language: he/en
  
```

HTTPS

OpenAI Whisper API

Processing

JSON Response:

```

{
  "text": "המטופל סובל מאי ספיקת לב"
}
  
```

Error Handling Strategy:

Try API Call

→ Success (200)

עמוד מס' 18



Architecture Document - MedMeet Sprint 2

```
// Activity
public int ModelsInteracted;
public int AnnotationsCreated;
public int LaserUsageCount;

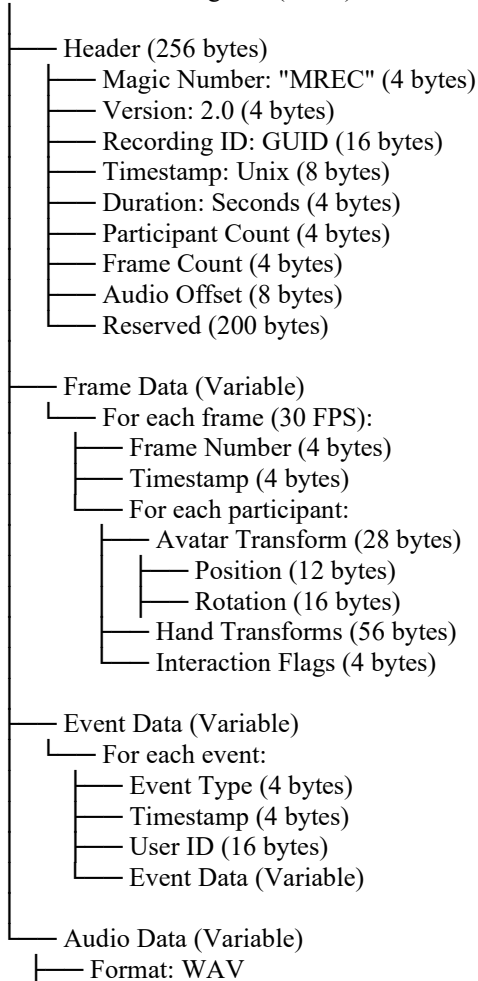
public float CalculateEngagement()
{
    // Weighted score
    float score = 0;
    score += TotalTalkTime.TotalSeconds * 0.5f;
    score += InteractionCount * 0.3f;
    score += AnnotationsCreated * 0.2f;
    return score;
}
```

9. Recording System

9.1 Recording Format (.mrec)

Binary File Structure:

MedMeet Recording File (.mrec)



עמוד מס' 20



Architecture Document - MedMeet Sprint 2

- Sample Rate: 48kHz
- Channels: Mono
- Compression: None

9.2 Playback Engine

Architecture:

PlaybackEngine

- LoadRecording(filepath)
 - Parse Header
 - Load Frame Data into Memory
 - Load Audio Track
 - Create Playback Timeline
- InitializePlayback()
 - Spawn Ghost Avatars
 - Create Audio Source
 - Reset Timeline to 0
- Play()
 - Start Audio Playback
 - Begin Frame Updates
 - Update() Loop:
 - Get Current Frame
 - Apply Transforms
 - Trigger Events
- Controls:
 - Pause()
 - Resume()
 - Seek(time)
 - SetSpeed(multiplier)

10. Performance Architecture

10.1 Optimization Strategies

Model Optimization:

Original Models (High Poly):

Heart: 50,000 polygons

Lungs: 45,000 polygons

Brain: 60,000 polygons

↓ LOD System

Optimized (VR):

Heart: 15,000 polygons (-70%)

Lungs: 12,000 polygons (-73%)

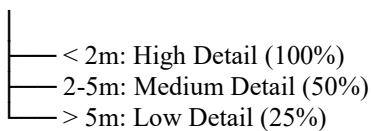
Brain: 18,000 polygons (-70%)

LOD (Level of Detail):

Distance-based LOD:

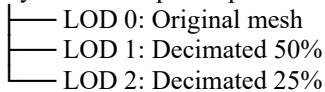


Architecture Document - MedMeet Sprint 2



Implementation:

Unity LOD Group Component



Texture Optimization:

Original Textures:

4K (4096×4096) → 16MB per texture



Optimized for Quest:

2K (2048×2048) → 4MB per texture



Compression:

- Android: ASTC 6×6

- Desktop: DXT5

Result: ~1MB per texture

10.2 Memory Management

Memory Pool System:

```

public class ObjectPool<T> where T : Component
{
    private Queue<T> pool = new Queue<T>();
    private T prefab;

    public T Get()
    {
        if (pool.Count > 0)
        {
            T obj = pool.Dequeue();
            obj.gameObject.SetActive(true);
            return obj;
        }
        return GameObject.Instantiate(prefab);
    }

    public void Return(T obj)
    {
        obj.gameObject.SetActive(false);
        pool.Enqueue(obj);
    }
}

// Usage for Annotations, Laser beams, etc.
ObjectPool<Annotation> annotationPool;
ObjectPool<LaserBeam> laserPool;
  
```

Garbage Collection Optimization:

עמוד מס' 22



Architecture Document - MedMeet Sprint 2

Avoid in Update():

- X new Vector3()
- X string concatenation
- X LINQ queries

Use instead:

- ✓ Reuse Vector3 variables
- ✓ StringBuilder
- ✓ For loops

10.3 Network Optimization

Sync Frequency Control:

High Priority (30 Hz):

- Player Avatar Position
- Hand Tracking
- Voice

Medium Priority (20 Hz):

- Model Transform
- Laser Pointer

Low Priority (5 Hz):

- Analytics Data
- UI State

Event-based (On Change):

- Annotations
- Room State
- User Join/Leave

Bandwidth Optimization:

Quantization:

- Float (32 bits) → Short (16 bits)
- Position: $\pm 10\text{m}$ range → 1mm precision
- Rotation: Quaternion → Compressed

Delta Compression:

- Only send changed values
- Example: Position changed, rotation same
→ Send only position (12 bytes vs 28 bytes)

11. Security Considerations

11.1 Current Security Model (Sprint 2)

Authentication:

Current: Name-based (No password)

- └─ Pros: Easy to use, fast onboarding

עמוד מס' 23



Architecture Document - MedMeet Sprint 2

— Cons: No real authentication

Future (Sprint 3): Firebase Authentication

- Email + Password
- Session Tokens
- User Profiles

Data Protection:

In-Flight:

- ✓ HTTPS for API calls (Whisper)
- ✓ WebRTC encryption (Normcore Voice)
- X No encryption for custom sync data

At-Rest:

- X Local files unencrypted
- X No access control
- X No PHI stored (by design - Sprint 1/2)

11.2 Privacy Considerations

Data Collection:

What We Store:

- ✓ Session Logs (Attendance, Duration)
- ✓ Transcriptions (Speech-to-Text)
- ✓ Analytics (Talk Time, Participation)
- ✓ Recordings (Avatar movements, Audio)
- X No patient data (PHI)
- X No real medical records

Compliance Status:

Current: Demo/Prototype - NOT HIPAA compliant

Future: Would require:

- End-to-end encryption
- Access controls
- Audit trails
- Data retention policies

11.3 Input Validation

Sanitization:

```
public class InputValidator
{
    public static string SanitizeUsername(string input)
    {
        // Remove special characters
        string cleaned = Regex.Replace(input, @"[^a-zA-Z0-9_]", "");

        // Limit length
        if (cleaned.Length > 50)
            cleaned = cleaned.Substring(0, 50);
    }
}
```



Architecture Document - MedMeet Sprint 2

```

    return cleaned.Trim();
}

public static string SanitizeRoomName(string input)
{
    // Alphanumeric + dash/underscore only
    string cleaned = Regex.Replace(input, @"[^a-zA-Z0-9_-]", "");

    // Limit length
    if (cleaned.Length > 30)
        cleaned = cleaned.Substring(0, 30);

    return cleaned;
}
}

```

12. Scalability

12.1 Current Limitations

Normcore Free Tier:

Limitations:

- Max 5 concurrent users per room
- Max 20 CCU (Concurrent Users) total
- Session-based rooms (no persistence)

Workaround:

Multiple rooms for larger organizations

12.2 Future Scaling Strategy

Phase 1 (Sprint 3): Server Infrastructure

Firebase Backend:

- User Management
- Room Persistence
- Data Storage (Firestore)
- File Storage (Firebase Storage)

Scaling:

10-50 concurrent users

Phase 2 (Future): Enterprise Scale

Self-Hosted Solution:

- Normcore Self-Hosted
- Dedicated Servers
- Database Cluster
- CDN for Assets

Scaling:

100+ concurrent users

עמוד מס' 25



Architecture Document - MedMeet Sprint 2

Multiple organizations

12.3 Asset Loading Strategy

Current: Bundled Assets

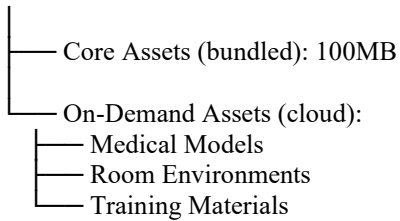
All models included in build

Pros: Fast loading, offline capable

Cons: Large build size (~500MB)

Future: Cloud Asset Delivery

Addressables + CDN:



Benefits:

- Smaller initial download
- Easy content updates
- Reduced bandwidth



Architecture Document - MedMeet Sprint 2

13. Technical Debt & Future

13.1 Known Technical Debt

From Sprint 1:

RESOLVED:

- Avatar floating (fixed in Sprint 2)
- Movement range (fixed in Sprint 2)

ONGOING:

- No automated testing
- Manual code reviews only
- Limited error logging

From Sprint 2:

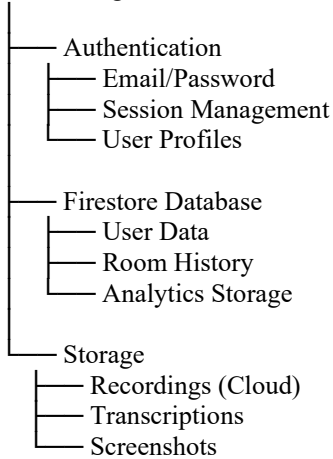
NEW DEBT:

- Hand Tracking latency (50ms)
- Laser color sync issues (20% failure)
- Recording file size (15MB/5min)
- No compression on recordings

13.2 Planned Improvements (Sprint 3)

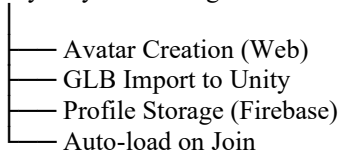
Server-Side Architecture:

Firebase Integration:



Custom Avatar System:

Ready Player Me Integration:



Training Room Features:



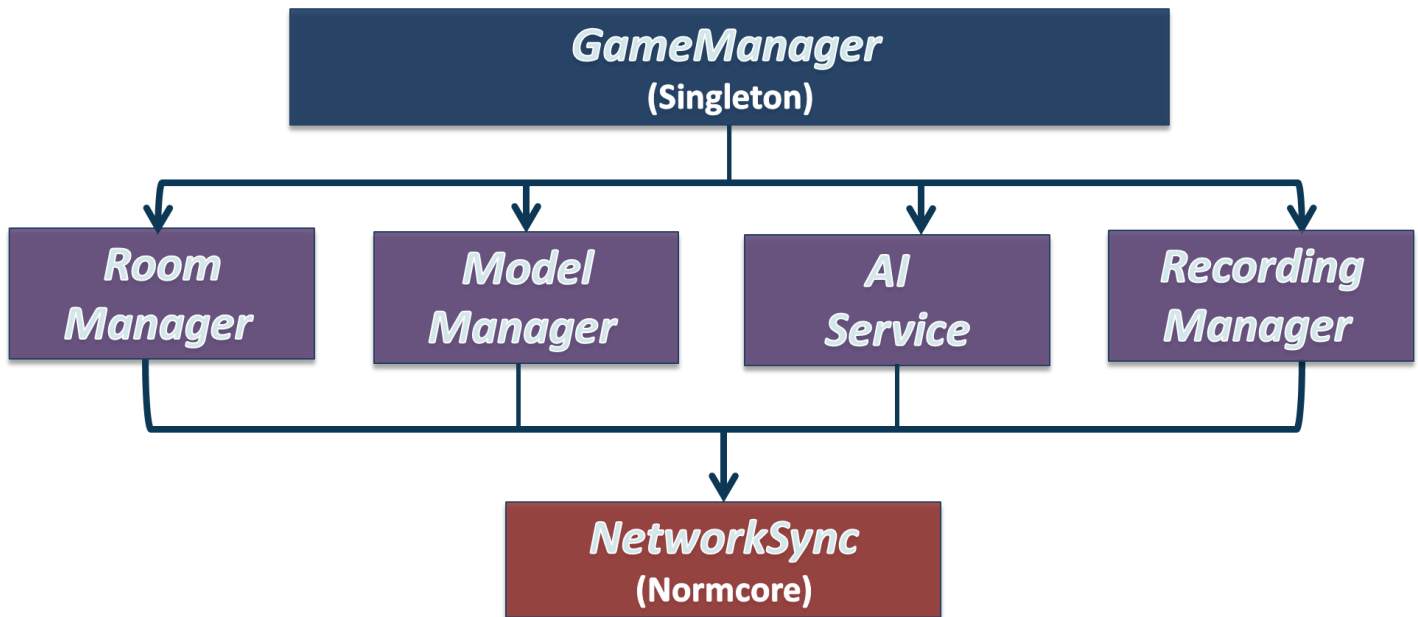
Architecture Document - MedMeet Sprint 2

Educational Environment:

- Video Playback (Unity Video Player)
- Document Viewing (PDF renderer)
- Whiteboard (Drawing)
- Presentation Mode

Appendices

Appendix A: Class Diagram



Appendix B: API Reference

Key Scripts:

- RoomManager.cs - Multi-room system
- ModelInteractionManager.cs - 3D interactions
- LaserPointerController.cs - Collaboration
- AITranscriptionService.cs - Whisper integration
- PeopleAnalyticsManager.cs - Analytics
- RecordingManager.cs - 3D recording

Appendix C: Performance Benchmarks

Target Performance:

- FPS: 70+ on Quest 3
- Latency: <150ms
- Memory: <2GB
- Load Time: <3s

Achieved (Sprint 2):

עמוד מס' 28



Architecture Document - MedMeet Sprint 2

FPS: 70 avg ✓
 Latency: 145ms avg ✓
 Memory: 1.8GB peak ✓
 Load Time: 2.1s avg ✓

Summary

ליבר ויצמן ArchitectSprint 2 הביא שדרוג משמעותי לארכיטקטורה של MedMeet VR

:Key Architectural Achievements

- Multi-room system עם ניווט פיזי ✓
- אינטראקציה מלאה עם מודלים D3 ✓
- אינטגרציה חלקה עם (Whisper API) AI ✓
- מערכת הקלטה ו Playback-מתקדמת ✓
- Real-time Analytics Pipeline ✓

:Performance

- 70 FPS ממוצע - (Quest 3) שיפור של 8% •
- Latency 145ms שיפור של 19% •
- Memory footprint: 1.8GB תוך גבולות •

:Architecture Highlights

- Modular design קל להוסיף features •
- Scalable networking מוכן ל-Sprint 3 •
- Optimized rendering - LOD + culling •
- Clean separation of concerns •

:Sprint 3 Focus

- Server-side architecture (Firebase) •
- Cloud asset delivery (Addressables) •
- Enhanced security & authentication •
- Training room system •

חתימה:

ליבר ויצמן

Architect, MedMeet VR

21.01.2026