Tumor Tracking Web Application – Workshop Project Document (HLD)

Project Overview

This project aims to streamline and improve clinical workflows by providing an automated system for analyzing tumor progression across multiple MRI scans. Currently, medical personnel must manually compare large volumes of imaging data, which is time-consuming, prone to error, and difficult given the complexity and variability of MRI studies. The system will generate an automated tumor-tracking report that summarizes detected changes across time, helping clinicians make more informed and precise decisions.

Problem Definition

Doctors and radiologists are required to interpret MRI scans from various time points to determine tumor trajectory. This process involves significant cognitive load and can lead to overlooked trends or delayed treatment decisions. A streamlined system for tumor progression analysis would reduce that burden and offer standardized output that supports clinical decision-making.

Objectives

- Build a full-stack web application that accepts MRI scans and generates automated tumor trajectory reports.
- Provide segmentation, multi-scan comparison, and structured PDF output.
- Support clinician workflows with patient history, user authentication, and secure storage.
- Implement privacy controls to ensure all medical data remains secure and compliant.

Key Features

Core Features

- **Tumor Segmentation:** Using nnU-Net to identify tumors within MRI scans.
- **Multi-Scan Tumor Tracking:** Automatic comparison across multiple MRI studies.
- **Automated PDF Reporting:** Summaries of tumor volume, morphological changes, and progression.
- **Integration of RAG-based patient data summaries**
- **Authentication System:** Doctors access only their assigned patient data.

Nice-to-Have Features

- Enhanced UI and dashboard components.
- Annotation tools for clinicians.
- 3D MRI visualization.
- Privacy system for regulations handling.

Target Audience

This system targets the following audience:

- **Primary users:** Doctors, especially oncologists and radiologists.
- **Secondary users:** Radiation therapy personnel and medical imaging staff.

User Stories

Oncologist:

- upload a patient's current MRI scan and have it automatically aligned with their scan from 3 months ago, so that he can visually verify if the tumor size has increased or decreased without manually scrolling through slices.
- Should have all his patients data, scans, ability to load new scans manually, and ability to initiate the analysis system if needed. Later, needs to have the ability to check the scans manually if need to verify the results, and see the decision making process.

Radiologist

- want the system to calculate the exact volumetric change (in cubic centimeters) of the tumor between two time points, so that he can report objective "Response to Treatment" metrics, rather than subjective estimates.
- Uploads new scans and initiate the automatic insight model system to generate the PDF report.

Clinician:

- want to see a summarized textual report that integrates the MRI findings with the patient’s recent general health data (age, symptoms, comorbidities), so that I have a holistic view of the patient's status before making a treatment decision.
- Will use the user data, and the reports from the system.

Medical Administrator:

- I want to ensure that only assigned doctors can view specific patient records via role-based authentication, so that we remain compliant with hospital privacy regulations.
- Will have an admin dashboard to assign patients to doctors, and mange permissions.

Existing Approaches and Our Differentiation

Existing Products

Companies such as Aidoc, Viz.ai, Arterys (Tempus), RadAI, and Qure.ai provide AI-driven radiology tools but focus on triage or single-scan interpretation rather than longitudinal tumor tracking.

Research tools like BraTS models, 3D Slicer, and academic segmentation papers offer strong segmentation but lack robust, automated multi-scan comparison and practical clinical integration.

Our Competitive Advantage

- **Longitudinal Analysis:** Automated tumor tracking across multiple MRI scans.
- **Standardized Reporting:** PDF summaries designed for clinical workflows.
- **Hospital-Focused Architecture:** Built directly from pain points reported by Ichilov Hospital personnel.
- **Secure Deployment:** Privacy-first design to meet regulatory definitions.

Existing Models and our utilization:

nnU-Net:

The system uses **nnU-Net** as the core tumor segmentation engine due to its strong performance, robustness, and widespread adoption in medical imaging research. nnU-Net is integrated as a **dedicated inference service** within the image processing pipeline and is executed asynchronously via worker nodes. The system **does not train models from scratch**; instead, it utilizes **pretrained nnU-Net models**, allowing reliable segmentation performance while significantly reducing computational cost, development time, and data requirements.

For inference, imaging data is converted to **NIfTI format** and organized in a lightweight, standardized directory structure compatible with nnU-Net's prediction interface. Each study or timepoint is placed in an input directory containing one or more NIfTI volumes with consistent naming conventions. An example inference input layout is shown below:

nnU-Net produces corresponding segmentation masks as output volumes, which are stored as derived artifacts and passed to downstream components for tumor volume calculation, longitudinal comparison, and clinical report generation. By treating nnU-Net as an isolated inference service, the system ensures modularity, reproducibility, and the ability to replace or upgrade segmentation models without impacting the surrounding infrastructure.

Medical Imaging Data Formats: DICOM and NIfTI

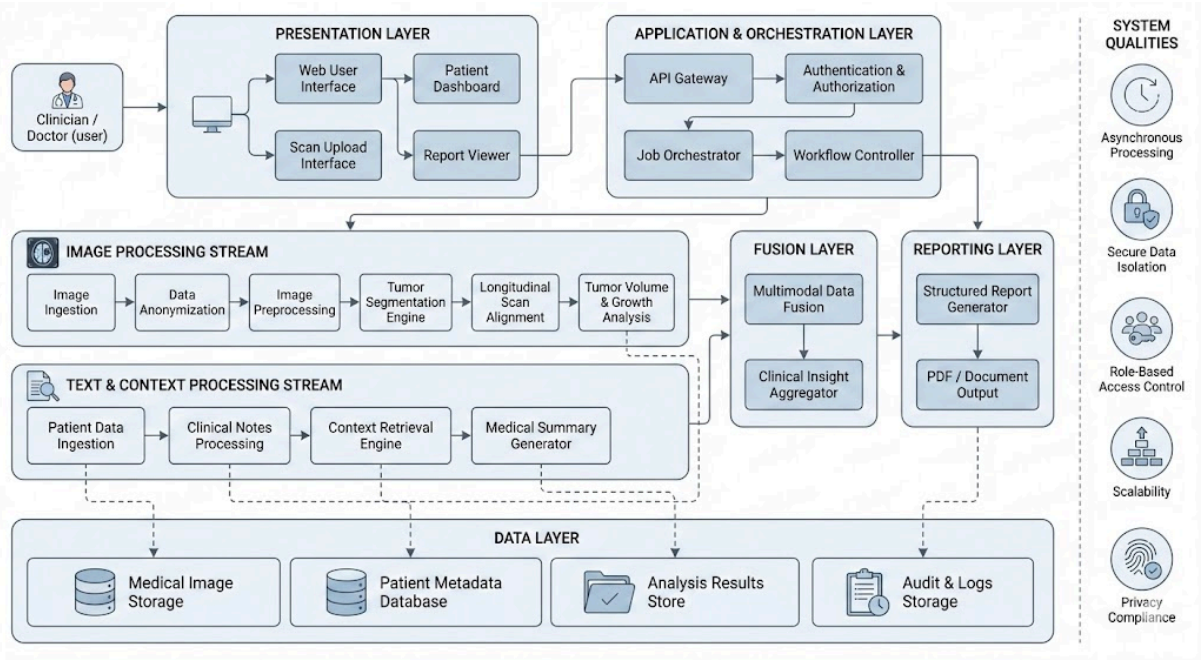
The system supports **DICOM** and **NIfTI (NIfTI-1)** formats, which are the dominant standards for medical imaging data. **DICOM** is used for clinical acquisition and storage and contains rich metadata describing the patient, study, imaging protocol, and scanner parameters. Upon ingestion, DICOM series may be converted into **NIfTI format**, which provides a compact, array-based representation of 3D medical volumes optimized for numerical processing.

NIfTI preserves essential spatial information such as voxel spacing and orientation while enabling efficient interaction with deep learning frameworks such as nnU-Net. Internally, all segmentation and quantitative analysis operations are performed on NIfTI volumes, while original DICOM files are retained for traceability and clinical reference. Supporting both formats allows the system to integrate smoothly with hospital imaging workflows while maintaining a standardized and performant internal representation for machine learning and longitudinal tumor analysis.

High-Level Solution Architecture

This system utilizes a **Dual-Stream Processing Architecture** to handle the distinct nature of image data versus textual patient data.

High-Level Input/Output Flow:



1. The Image Stream (Tumor Detection):

- **Input:** Longitudinal MRI sequences (DICOM/NIfTI). Fetching data from local data sources by client (data don't leave the hospital)
- **Processing:** The system uses **Deep Learning (for example: nnU-Net)** for segmentation. It identifies the tumor in 3D space and performs geometric comparisons between Timepoint A and Timepoint B.

- **Customization:** The model adapts to the specific anatomical features of the patient's scan to generate a precise mask.

2. The Text Stream (Patient Context via RAG):

- **Input:** Unstructured text (Medical history, pathology reports, physician notes) and structured data (Age, Gender).
- **Technique (RAG vs. Fine-Tuning):** We utilize **Retrieval-Augmented Generation (RAG)** rather than fine-tuning an LLM.
    - **Why RAG?** Patient data is dynamic and highly specific. Fine-tuning a model on every patient is computationally expensive and risks "hallucination." RAG allows us to retrieve exact facts (e.g., "Patient had surgery on date X") from the specific patient's documents to prompt a frozen LLM, ensuring accuracy and data isolation.
- **Processing:** Clinical notes are chunked and stored in a vector database. When generating the report, the system queries this database to summarize the patient's general health status alongside the imaging results.

Data Sources and Training Strategy

To build a robust solution, we will access data from the following sources:

- **Ichilov tumor dataset:** Data set provided by Ichilov for scientific research.
- **nnU-Net (Segmentation Core):** Self-adapting Framework for U-Net-Based Medical Image Segmentation – updates every year with BraTS challenge.
- **BraTS Dataset:** Updated every year for the BraTS challenge.

System Architecture and Technologies

The system follows a service-oriented, cloud-ready architecture with clear separation between frontend, backend, processing pipelines, and storage.

Main architectural layers:

- Presentation Layer (Web UI)
- Application Layer (API & orchestration)
- Processing Layer (ML & NLP pipelines)
- Data Layer (databases and object storage)

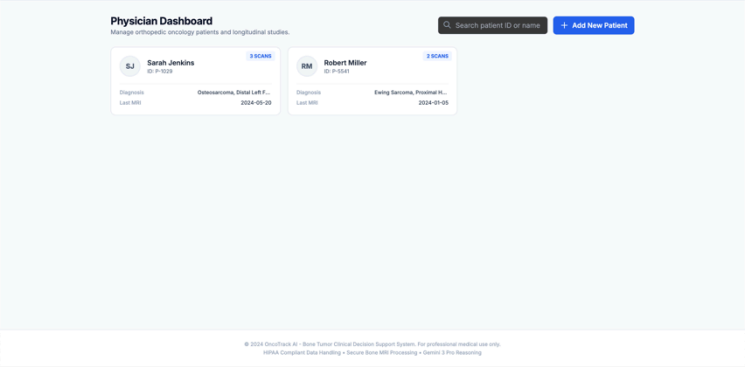
Multi-agent layer across all layer to interact with the data and users

Frontend

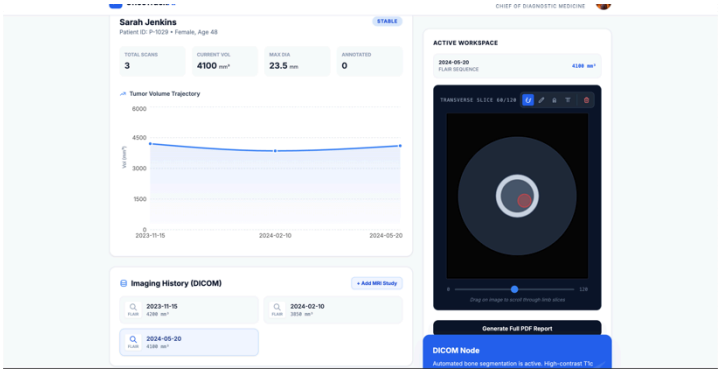
- Using React.js technology with Typescript.

Example for UI functionality:

Dr. patients dashboard:



Patient page:



Backend

- **Languages & Frameworks:** Python, FastAPI
- **ML Frameworks:** PyTorch, nnU-Net. (consider other technologies such as MONAI, and TorchIO)
- **Imaging Libraries:** optional technologies are Pydicom, SimpleITK, and NiBabel
- **Task Processing:** Celery with Redis
- **Database:** PostgreSQL
- **Reporting:** PDF generation service

Infrastructure

- **Cloud:** AWS (EC2, S3, RDS, VPC, ECR, CloudWatch, KMS)
- **Containerization:** Docker, Docker Compose
- **CI/CD:** GitHub Actions

- **Security:** Private VPC, encryption in transit and at rest, audit logs (read and write), 2FA, role-based control access.

Architecture Style

- Asynchronous Service-Oriented Architecture: inference service, API gateway, frontend, worker service, database, storage.

Multi-Agent System Integration (Conceptual Design)

A multi-agent system can be naturally integrated into the project to orchestrate and optimize complex decision-making processes across the pipeline. In this context, each agent represents a specialized responsibility, such as **image preprocessing**, **segmentation execution**, **longitudinal comparison**, **clinical context summarization**, or **report validation**. These agents operate semi-independently while communicating through well-defined interfaces and shared system state.

Such an approach enables flexible pipeline composition, adaptive execution strategies, and clearer separation of concerns. For example, a coordination agent may decide which segmentation strategy to apply based on image quality or modality, while an evaluation agent can validate outputs before report generation. Integrating a multi-agent design supports future experimentation, improves system robustness, and aligns with the project's research-oriented nature by enabling comparative evaluation of alternative processing strategies.

Scalability and Performance Considerations

The system is designed with scalability in mind to support increasing numbers of patients, imaging studies, and concurrent users without degradation in performance. Scalability is primarily achieved through a decoupled, service-oriented architecture, where computationally intensive tasks such as image segmentation, registration, and report generation are executed asynchronously by worker services. This prevents long-running operations from blocking the main application flow and ensures responsiveness of the user interface.

Performance-critical components, particularly nnU-Net inference, are isolated into dedicated worker pools that can be scaled horizontally based on workload and available GPU resources. The backend API layer is stateless, enabling replication behind a load balancer if required. Data access patterns are optimized by separating raw imaging data (stored in object storage) from metadata and job state (stored in a relational database), reducing database load during heavy processing periods. These design choices allow the system to scale incrementally and adapt to both academic-scale experiments and larger clinical deployments.

Error Handling and Logging

Robust error handling and logging mechanisms are essential due to the complexity of medical data processing pipelines and the sensitivity of clinical workflows. The system implements job-level error isolation, ensuring that failures in processing a specific study do not affect other jobs or system components. Each asynchronous task transitions through well-defined states (e.g., pending, running, completed, failed), allowing precise identification of failure points.

Errors are logged in a structured manner, capturing contextual information such as job identifiers, patient references (non-identifying), processing stage, and timestamps. This enables effective debugging, monitoring, and post-mortem analysis. User-facing error messages are intentionally high-level and non-technical, informing clinicians of failure states without exposing internal system details. Detailed logs are reserved for developers and system administrators, supporting maintainability and continuous system improvement.

Constraints and Dependencies

- MRI training data is limited and expensive.
- GPU resources are required for training and inference.
- Strict privacy constraints due to handling medical data (PHI).
- All patient data must remain fully secure and private.

Challenges and Risks

Technical Challenges

A central technical challenge of the system is **reliably matching and tracking the same tumor across multiple MRI scans acquired at different timepoints**. Variations in patient positioning, scanner hardware, and acquisition parameters introduce inconsistencies that complicate longitudinal comparison. Additionally, the system must remain robust to differences in imaging protocols and scan quality, which are common in real-world clinical environments.

Another significant challenge is the **efficient processing of large 3D volumetric datasets**. MRI volumes are computationally intensive in terms of memory usage and processing time, particularly when segmentation and registration operations are applied across multiple timepoints. Ensuring acceptable performance while maintaining accuracy is therefore a key technical concern.

Design Challenges

The system must present **complex medical and computational outputs**—including volumetric measurements, temporal changes, and segmentation-derived insights—in a manner that is intuitive and interpretable for clinicians. Overly technical representations risk misinterpretation, while excessive simplification may obscure important clinical details. Achieving an appropriate balance between clarity and informational depth is a key design challenge.

Algorithmic Challenges

Accurate tumor segmentation across timepoints presents inherent algorithmic challenges. Even when using a pretrained segmentation model, inconsistencies may arise due to imaging noise, anatomical changes, or model sensitivity to scan variations. These inconsistencies can propagate into downstream tumor comparison and growth assessment.

Image registration across heterogeneous scans is an additional challenge. Misalignment between scans can lead to inaccurate spatial comparison and misleading tumor growth metrics. Selecting and configuring appropriate registration strategies is therefore critical to ensuring meaningful longitudinal analysis.

Security and Privacy Assurance

1. Security Expectations and Rationale

Our system design prioritizes realistic, industry-standard defenses to protect Protected Health Information (PHI). We adhere to following core security pillars:

Security Pillar	Operational Expectation	Risk Rationale (The "Why")
Patient Privacy & Data Isolation	**Complete De-identification & Logical Segregation:** The system must ensure that no Direct Identifiers (PII/PHI) persist in the processing pipeline. Furthermore, user sessions (especially in the RAG module) must be strictly isolated to prevent cross-patient data leakage.	**Regulatory Compliance & Leakage Prevention:**Adherence to Israeli MOH Circular 02/2021 and GDPR requirements. This mitigates the risk of **Re-identification Attacks** (linking metadata back to a person) and prevents "Hallucination Leakage" where an AI model accidentally exposes one patient's data to another doctor.

Data Integrity & Non-Repudiation	**Bit-Exact Fidelity & Immutable Reporting:** The system must guarantee that the MRI analyzed is bit-for-bit identical to the source upload. Final clinical reports must be cryptographically signed to prevent post-generation alteration.	**Mitigation of Tampering & Corruption:** Prevents **Man-in-the-Middle (MitM)** attacks where data is altered during transit. It protects against malicious internal actors changing tumor metrics in the report, ensuring clinicians can trust the output is authentic and unmodified.
Zero Trust Access Control	**Resource-Aware Authorization:**Authentication alone is insufficient. The system must enforce "Need-to-Know" policies at the object level, verifying that the requesting user has an explicit, active relationship with the specific patient record being requested.	**Prevention of Horizontal Privilege Escalation:**Blocks the most common internal threat vector where valid users (e.g., Doctor A) access unauthorized records (e.g., Doctor B's patients). This minimizes the internal attack surface and ensures privacy even among authorized staff.
Network Containment & Perimeter Defense	**Public Isolation of Data Core:** The persistence layer (Database, Storage) and Inference Engine must operate within a "Dark Network" (Private Subnet), completely unreachable from the public internet.	**Blast Radius Reduction:** Ensures that even in a worst-case scenario where the public-facing Frontend or API Gateway is compromised, the attacker cannot directly access or exfiltrate the raw database or imaging archives.

2. Architectural Integration of Security Solutions

We have selected implementations that are robust yet feasible within our project timeline:

Security Domain	Solution Strategy	Technical Implementation	Architectural Integration
Data De-identification & Anonymization	Automated DICOM Attribute Sanitization & Pixel Scrubbing	Utilization of pydicom library to systematically strip VR=PN/LO/DA tags upon ingestion. Patient IDs are replaced with internal UUIDs (Pseudonymization). Facial features in brain MRIs are obfuscated via PyDeFace (Best-effort fallback to metadata-only if segmentation risk detected).	**Pre-Ingestion Middleware:** Executed within an isolated worker container immediately after file upload and prior to persistence in S3 or Database.
Data Integrity & Non-Repudiation	Cryptographic Hashing & Digital Signing	**Input:** SHA-256 hash generation upon file receipt to detect corruption or tampering. **Output:** Digital signing of generated PDF reports to create an immutable audit record.	**API Gateway & Worker:** Hash verification occurs at the entry gate (API) and is re-verified by the Inference Engine. Signing is embedded in the Report Generation Microservice.
(optional) Generative AI Isolation (RAG)	Ephemeral Context Windows & Tenant Segregation	Implementation of non-persistent vector indices. Patient-specific context is loaded into a temporary in-memory store (e.g., Redis Vector / Faiss) for the active session only	**Chat Service Orchestrator:** Managed by the backend session manager; ensures strict logical separation where Query N cannot access Context N-1.

		and flushed immediately post-interaction.	
Access Control & Authorization	Attribute-Based Access Control (ABAC) / Fine-Grained RBAC	Custom Middleware enforcing "Resource Ownership" policies. Validates that the requesting DoctorID has an explicit assignment relationship to the target PatientID in the relational schema.	**API Middleware Layer:** Intercepts every HTTP request before reaching the business logic controllers, returning 403 Forbidden on unauthorized resource access attempts.
Forensic Observability (Logging)	Tamper-Proof Audit Trails	Centralized logging of all "Create, Read, Update, Delete" (CRUD) operations. Logs capture Actor, Action, Timestamp, and Resource ID without revealing PHI (Private Health Information).	**Sidecar Logging Service:** Asynchronous log collector pushing events to a secured, write-only S3 bucket or CloudWatch stream, distinct from application storage.
Network Micro-Segmentation	Private VPC & Subnet Isolation	Deployment within a strict AWS VPC topology. Public ingress is restricted solely to the API Gateway / Load Balancer. Database and Inference nodes reside in private subnets with no internet gateway.	**Infrastructure Layer:** Enforced via AWS Security Groups and ACLs, creating a DMZ for the frontend and a "Dark Network" for the data core.
End-to-End Cryptographic Protection	AES-256 Storage & TLS 1.3 Transit Encryption	**At Rest:** AWS KMS managed keys encrypting all RDS instances and S3 buckets. **In Transit:** Enforced HTTPS/TLS 1.3 with strong cipher suites for all client-server and	**Platform-Wide:** Transparently handled by the cloud provider's infrastructure (AWS) and application-level SSL context configurations.

		inter-service communication.	
(optional) Identity Assurance	Multi-Factor Authentication (MFA) & Secure Session Management	Enforcement of 2FA for all doctor logins. Use of short-lived, signed JWTs (JSON Web Tokens) for session validation to prevent session hijacking.	**Authentication Service:**Integrated with the Identity Provider (IdP) at the login gateway; JWT validation occurs at every API endpoint call.
(optional) Supply Chain Security	Container Image Scanning & Dependency Auditing	Automated CI/CD pipeline steps that scan Docker images and Python requirements.txtfor known CVEs (Common Vulnerabilities and Exposures) before deployment.	**CI/CD Pipeline (GitHub Actions):** Blocking gate that prevents deployment of artifacts containing high-severity vulnerabilities.

Expected Impact

The proposed system is expected to improve the **consistency and reliability of longitudinal tumor monitoring** by providing a structured and repeatable pipeline for comparing MRI scans across timepoints. By standardizing the analysis process, the system reduces variability introduced by manual measurements and subjective interpretation, leading to more stable and comparable assessments of tumor progression.

In addition, the system aims to **reduce the manual workload for radiologists and oncologists** by automating time-consuming tasks such as tumor segmentation, volumetric measurement, and report preparation. This automation allows clinicians to focus on higher-level clinical decision-making rather than repetitive technical analysis.

Finally, by enabling systematic comparison of tumor metrics and trends over time, the system supports **earlier identification of clinically meaningful tumor changes**. Early detection of growth or response patterns may contribute to more timely treatment adjustments and improved patient management.

Future Extensions

The proposed architecture is intentionally extensible to support future research directions and feature enhancements. Potential extensions include support for **additional imaging modalities** (e.g., CT or PET), **multi-lesion or multi-organ tracking**, and the integration of advanced visualization tools for improved interpretability of segmentation results. The modular inference design also allows replacement or augmentation of the segmentation engine as newer models or methods become available.

From a research perspective, future extensions may include personalization of analysis pipelines, incorporation of temporal modeling techniques, and exploration of explainable AI methods to improve clinician trust. On the system level, collaboration features, cross-institutional deployments, and integration with hospital information systems (HIS/PACS) are considered viable long-term directions.

Relevant Paper References

To ensure our methodology is grounded in state-of-the-art research, we are referencing the following key papers:

1. **nnU-Net (Segmentation Core):** Self-adapting Framework for U-Net-Based Medical Image Segmentation (2018).
2. **The 2024 Brain Tumor Segmentation (BraTS) Challenge:** Glioma Segmentation on Post-treatment MRI
3. **Automated longitudinal treatment response assessment of brain tumors:** A systematic review (2025)
4. **Retrieval-Augmented Generation (RAG) in Healthcare:** A Comprehensive Review (2025)
5. **Artificial Intelligence-Enabled Medical Devices** (website)