Contents

1 Segment Platform — Research Foundation Document

Browser-Based Interactive 3D Medical Segmentation with SAM-Med3D-turbo

Field	Detail
Institution	Charité – Universitätsmedizin Berlin
Core Stack	TypeScript · Vite · Niivue 0.66 · FastAPI · PyTorch
Foundation Model	SAM-Med3D-turbo (3D ViT-B/16, 91 M params)
Primary Dataset	DISCHARGE (25 M CCTA images, 3 561 patients)

Field	Detail
Pre-training Corpus	SA-Med3D-140K (21 729 volumes, 143 518 masks, 245 categories)
Clinical Domains	Coronary CTA (lumen / outer wall / plaque) · Prostate mpMRI (zones / lesions / OARs)
Document Version	2026-02-18

1.1 Table of Contents

1.2 1. Executive Summary & Research Vision

1.2.1 1.1 Clinical Gap

Problem	Impact
Manual coronary segmentation	30–60 min / case, €200–400
Limited scalability	DISCHARGE (3 561 patients) still largely un-segmented
Prostate mpMRI zone/lesion contouring	Equally time-intensive for biopsy/radiotherapy planning
Centre-specific expertise	Manual segmentation available only at specialised sites

1.2.2 1.2 Our Solution

A **single, universal, browser-based platform** powered by **SAM-Med3D-turbo**:

- **1–3 point prompts** (3D coordinates in mm space) any anatomy, any modality
- **Zero installation** — runs in Chrome / Firefox / Safari
- **On-premise GPU inference** at Charité (GDPR-compliant; no data leaves the hospital)
- **Interactive workflow** for radiologists + **batch research mode** for DISCHARGE processing + **active-learning loop** for continuous improvement

1.2.3 1.3 Research Foundation

This platform is a **test-bed for foundation-model research** in cardiovascular imaging:

1. Evaluate **zero-shot / few-shot generalisation** of SAM-Med3D-turbo on unseen DISCHARGE cases
2. Quantify **active-learning gains** (weekly fine-tuning on expert corrections)
3. Benchmark against **nnU-Net** task-specific models on clinical endpoints (stenosis %, plaque burden, FFR-CT correlation)
4. Open-source modular components for the broader MedAI community

1.2.4 1.4 Success Targets (6-month horizon)

Metric	Target
DISCHARGE cases auto-processed	> 80 % of dataset
End-to-end latency	< 2 s per vessel / per prostate gland
Dice (coronary lumen)	> 0.85 vs. expert
Dice (prostate whole-gland)	> 0.90 vs. expert
Cost reduction	10× cheaper than manual contouring

Critical note on Dice targets: The SAM-Med3D paper reports **87.12 % Dice on cardiac structures** with 1 prompt point (Table 5 in [1]). However, this was measured on the ACDC dataset (cardiac MRI short-axis cine), **not** coronary CTA. Coronary arteries are smaller, noisier, and motion-affected — published coronary lumen Dice values for task-specific models (nnU-Net) range 0.75–0.88 depending on vessel branch. Our 0.85 target is therefore ambitious but grounded.

1.3 2. Foundation Model: SAM-Med3D-turbo — Verified Technical Profile

1.3.1 2.1 Paper & Publication

Field	Value
Title	SAM-Med3D: Towards General-purpose Segmentation Models for Volumetric Medical Images
Authors	Haoyu Wang, Sizheng Guo, Jin Ye, Zhongying Deng, Junlong Cheng, Tianbin Li, Jianpin Chen, Yanzhou Su, Ziyan Huang, Yiqing Shen, Bin Fu, Shaoting Zhang, Junjun He, Yu Qiao
Venue	**ECCV BIC 2024 — Oral**
arXiv	2310.15161
License	Apache 2.0

1.3.2 2.2 Architecture (Verified from Paper §4.1 & GitHub)

SAM-Med3D uses a **fully 3D architecture trained from scratch** (Method 3 in the paper). The authors explicitly compared three adaptation strategies in preliminary experiments (Table 2):

Strategy	Seen Dice	Unseen Dice	Chosen?
3D Adapter + Frozen SAM	Lower	Moderate	□
Fine-tune SAM 2D□3D weights	Good	Poor (broken priors)	□
Train 3D from scratch	**Good**	**Best**	□

Rationale (Paper §4.1): "Training from scratch emerges as a better trade-off, exhibiting superior average performance" — the 2D-to-3D weight transition "might further break down the prior knowledge of SAM, which is harmful to generalization."

Component breakdown:

Component	Architecture	Parameters
Image Encoder	3D ViT-B/16 (3D positional encoding, 3D convolutions, 3D LayerNorm)	~86 M
Prompt Encoder	3D point/box encoder (learned embeddings)	~1 M
Mask Decoder	Lightweight 3D decoder (2-layer transformer + upsampling)	~4 M
Total		**~91 M**

Critical note: The paper states "86M encoder + 5M decoder" in various summaries. The exact split varies by source. The model is **not** a modified SAM (Meta) — it is architecturally distinct, sharing only the conceptual prompt-based paradigm.

1.3.3 2.3 Training (Verified from Paper §4.2)

Two-stage procedure:

Stage	Data	Epochs	Purpose
Stage 1: Pre-training	All 131 K masks from SA-Med3D-140K training set	800	Build general 3D medical understanding
Stage 2: Fine-tuning	~75 K filtered high-quality masks	Additional	Improve prompt efficiency on challenging targets

SAM-Med3D-turbo (from GitHub issue #2): - Fine-tuned on **44 additional datasets** beyond the base SA-Med3D-140K - Optimised for **sub-second inference** with FP16 - This is the recommended checkpoint for deployment

1.3.4 2.4 SA-Med3D-140K Dataset (Verified from HuggingFace)

Statistic	Value
Total 3D images	21 729
Total 3D masks	143 518
Anatomical categories	245
Modalities	28 (CT, MR, US, and more)
Sources	70 public datasets + 8 128 privately licensed cases from 24 hospitals
Primary task	General-purpose promptable segmentation

1.3.5 2.5 Verified Performance (from Paper Tables 3–5)

Overall (Table 3): - SAM-Med3D with 1 point: **+60.12 % overall Dice** improvement over original SAM - Consistently outperforms SAM-Med2D across all prompt counts - Operates at **1–26 % of inference time** compared to slice-by-slice SAM

By anatomy (Table 5, 1 prompt point):

Anatomy	SAM	SAM-Med2D	SAM-Med3D
Cardiac (seen)	Poor	Moderate	**87.12 %**
Organs (seen)	Poor	Moderate	**Best**
Lesions (unseen)	Very poor	Moderate	Competitive
Bones/muscles	Very poor	Moderate	**Greatest advantage**

Key finding (Paper §5.1.5): "SAM-Med3D using 1 point outperforms SAM-Med2D with N points in **45 targets out of 49**, achieving up to +68.2% improvement."

Transferability (Paper §5.1.6, Table 6): When SAM-Med3D's ViT encoder is used as pre-trained backbone for UNETR, it yields up to **+5.63 % Dice improvement** over training from scratch — confirming value as a foundation model.

> **Critical caveat for our project:** The "cardiac" results (87.12 %) are from the **ACDC dataset** (cardiac MRI, short-axis cine — segmenting LV/RV/myocardium). This is **not** coronary artery segmentation from CTA. Coronary arteries are 1.5–4 mm diameter, motion-affected, and require dual-wall segmentation — a substantially harder task not directly evaluated in the paper. The model has **never been benchmarked on coronary CTA lumen segmentation**. Our project will provide this missing evaluation.

1.3.6 2.6 Official Resources

Resource	Link
GitHub	uni-medical/SAM-Med3D
Paper	arXiv:2310.15161
Supplementary	ECCV Supplementary PDF
Turbo checkpoint	HuggingFace: sam_med3d_turbo.pth
Dataset	HuggingFace: SA-Med3D-140K
MedIM loader	uni-medical/MedIM
CVPR25 Challenge	MedSegFM Competition

1.3.7 2.7 Environment Setup (Verified from GitHub README)

```
conda create --name sammed3d python=3.10
conda activate sammed3d
pip install uv
uv pip install torch==2.6.0 torchvision==0.21.0 torchaudio==2.6.0
uv pip install torchio opencv-python-headless matplotlib prefetch_generator mona
```

1.3.8 2.8 Model Loading (Verified from GitHub)

```
import medim

# Option A: Direct from HuggingFace (downloads automatically)
ckpt_path = "https://huggingface.co/blueyo0/SAM-Med3D/blob/main/sam_med3d_turbo.
model = medim.create_model("SAM-Med3D", pretrained=True, checkpoint_path=ckpt_pa

# Option B: Local checkpoint (recommended for deployment)
model = medim.create_model(
    "SAM-Med3D",
    pretrained=True,
    checkpoint_path="app/models/checkpoints/sam_med3d_turbo.pth"
)

# Optimise for inference
import torch
device = "cuda" if torch.cuda.is_available() else "cpu"
model = model.to(device).half()  # FP16 for speed + memory
model.eval()
```

1.3.9 2.9 Data Format for Fine-tuning (Verified from GitHub)

SAM-Med3D expects **nnU-Net-style** directory layout with **binary masks**:

```
data/medical_preprocessed/
├── coronary/
│   ├── ct_DISCHARGE/
│   │   ├── imagesTr/
│   │   │   ├── discharge_0001.nii.gz
│   │   │   └── ...
│   │   └── labelsTr/
│   │       ├── discharge_0001.nii.gz  (binary mask)
│   │       └── ...
└── prostate/
    └── mr_CHARITE/
        ├── imagesTr/
        └── labelsTr/
```

Important (from GitHub): Ground-truth labels are required to generate prompt points during training. For inference without ground truth, "generate a fake ground-truth with the target region for prompt annotated."

1.3.10 2.10 Turbo vs. Standard Comparison

Metric	Standard (base .pth)	Turbo (sam_med3d_turbo.pth)
Pre-training data	131 K masks	131 K + 44 additional datasets
Average Dice (1 prompt)	~0.75	~0.82+
Inference time	4–8 s	0.5–1.5 s
VRAM usage	~10 GB (FP32)	~4 GB (FP16)

1.4 3. Clinical Domain A: Coronary CTA Segmentation (DISCHARGE)

1.4.1 3.1 DISCHARGE Trial Overview

Field	Value
Full name	Diagnostic Imaging Strategies for Patients with Stable Chest Pain and Intermediate Risk of Coronary Artery Disease
Reference	Dewey et al., NEJM 2022
Design	Multicentre randomised controlled trial
Patients	3 561
Images	~25 M CCTA images
Modality	Coronary Computed Tomography Angiography (CCTA)
Institution	Charité, Berlin (lead site)

1.4.2 3.2 Standardised Nomenclature (CAD-RADS / AHA 17-segment compatible)

Clinical Feature	Segmentation Class Label	Description	HU Range (contrast-enhanced CT)
Myocardium (LV)	`LV_MYO`	Muscular wall of left ventricle	50–120 HU
Endocardial Lumen	`Endocardial_Lumen`	Inner chamber blood pool	250–400 HU
Coronary Lumen	`Lumen_{LAD|RCA|LCx|LM}`	Vessel blood pool — per AHA branch	300–400 HU

Clinical Feature	Segmentation Class Label	Description	HU Range (contrast-enhanced CT)
Outer Wall (EEM)	`VesselWall_EEM`	External elastic membrane boundary	— (morphological)
Calcified Plaque	`Plaque_Calc`	High-density stable plaque	> 130 HU (often 500–1000)
Fibrous Plaque	`Plaque_Fibrous`	Intermediate-density plaque	60–130 HU
Low-Attenuation Plaque	`Plaque_LAP`	Lipid-rich / necrotic core — **high risk**	< 60 HU

1.4.3 3.3 Why Coronary Arteries are the Hardest Segmentation Target

Challenge	Detail	Impact
Motion artefacts	Heart beats 60–80 bpm; RCA most affected	Blurred vessel edges despite ECG gating
Small calibre	1.5–4 mm diameter	Partial volume effects at 0.5 mm resolution
Low-contrast plaque	Lipid-rich plaque 20–50 HU vs. lumen 300–400 HU	Nearly invisible on standard windowing
Blooming	Calcification > 130 HU causes beam hardening	Obscures adjacent soft plaque
Complex topology	Bifurcations, overlapping branches, tortuosity	Single-click prompts often insufficient
Dual-wall requirement	Must segment lumen AND outer wall separately	Wall thickness 0.5–3 mm = 1–5 voxels

1.4.4 3.4 SAM-Med3D-turbo Prompting Strategy for Coronary CTA

1.4.4.1 A. Multi-Point Centerline Prompt ("String" Prompt) Standard single-click prompts fail for thin, tortuous vessels spanning 100+ slices. Instead:

1. User clicks **proximal ostium** (positive point)
2. User clicks **distal vessel tip** (positive point)
3. Model "fills in" the lumen tube between points
4. If mask leaks into **great cardiac vein** → place **negative point** on vein

```
# Coronary lumen segmentation with multi-point prompt
lumen_mask = model.segment(
    volume=cta_volume,
    points=[ostium_xyz, mid_vessel_xyz, distal_xyz],
    labels=[1, 1, 1],  # all positive
)

# If leakage detected → add negative point
lumen_mask_refined = model.segment(
    volume=cta_volume,
    points=[ostium_xyz, mid_vessel_xyz, distal_xyz, vein_xyz],
    labels=[1, 1, 1, 0],  # last point is negative
)
```

1.4.4.2 B. Dual-Wall Nested Segmentation (Lumen → EEM)

Critical for calculating **stenosis %** and **plaque burden**:

1. **Step 1:** Segment lumen (bright contrast, easier target)
2. **Step 2:** Use lumen mask as **dense prompt** → expand outward to EEM
3. **Result:** `vessel_wall = outer_mask & ~lumen_mask` → plaque volume

```
# Step 1: Lumen
lumen_mask = model.segment(volume, points=[ostium, distal], labels=[1, 1])

# Step 2: Outer wall using lumen as prior
outer_mask = model.segment(
    volume,
    points=[ostium],
    dense_prompt=lumen_mask,  # prior mask guides expansion
    labels=[1]
)

# Step 3: Derive vessel wall
vessel_wall = outer_mask & ~lumen_mask
```

> **Critical note:** SAM-Med3D does **not** natively support a `dense_prompt` argument in its published codebase. The mask-as-prior strategy requires either (a) custom modification of the prompt encoder, or (b) a two-stage pipeline where the lumen mask is converted to additional point prompts along its surface. This is a research contribution we must implement.

```python
def characterise_plaque(volume, vessel_wall_mask):
    """Classify plaque components by HU value within the vessel wall mask."""
    hu_values = volume[vessel_wall_mask > 0]

    calcified    = (hu_values > 130).sum()
    fibrous      = ((hu_values >= 60) & (hu_values <= 130)).sum()
    lipid_rich   = (hu_values < 60).sum()
    total        = vessel_wall_mask.sum()

    return {
        "calcified_pct": calcified / total * 100,
        "fibrous_pct":   fibrous / total * 100,
        "lipid_rich_pct": lipid_rich / total * 100,
        "high_risk": (lipid_rich / total) > 0.04,  # >4% LAP = vulnerable
    }
```

1.4.4.3 C. Post-Processing: Plaque Characterisation by HU Thresholds

1.4.5 3.5 DISCHARGE-Specific Processing Considerations

Consideration	Detail
Scanner heterogeneity	Multi-centre trial varying scanner vendors, protocols, contrast timing
Reconstruction kernels	Soft vs. sharp kernels affect HU accuracy for plaque
Contrast timing	Early arterial phase optimal; late phase reduces lumen-wall contrast
ECG gating	Prospective vs. retrospective gating affects motion artefact severity
Data format	DICOM (clinical) convert to NIfTI.gz for model input
Annotations	MEDIS QAngio CT (expert contours) available for subset ground truth

1.5 4. Clinical Domain B: Prostate mpMRI Segmentation

1.5.1 4.1 Imaging Standard

Multiparametric MRI (mpMRI) is the clinical standard for prostate imaging, using: - **T2-weighted (T2W):** Anatomical detail — zonal anatomy - **Diffusion-weighted imaging (DWI) + ADC map:** Cellularity — lesion detection - **Dynamic contrast-enhanced (DCE):** Vascularity — supplementary

Reporting follows **PI-RADS v2.1** (Prostate Imaging-Reporting and Data System).

1.5.2 4.2 Anatomical Segmentation — The "Zones"

The prostate is divided into distinct zones with different MRI appearances and cancer risk:

Zone	Abbreviation	Cancer Risk	T2W Appearance	Clinical Role
Peripheral Zone	PZ	70–75 % of cancers	Bright (high signal)	Primary cancer surveillance region
Transition Zone	TZ	20–25 % of cancers	Heterogeneous (BPH nodules)	BPH assessment, cancer in older men
Central Zone	CZ	< 5 % of cancers	Low signal (dense stroma)	Rarely targeted; anatomical landmark
Anterior Fibromuscular Stroma	AFMS	Non-glandular	Very low signal	Can be invaded by anterior tumours

1.5.3 4.3 Pathology Segmentation — The "Lesions"

When segmenting pathology, the target is **clinically significant prostate cancer (csPCa)**:

Lesion Type	Description	Clinical Significance
Index Lesion	Largest / most aggressive tumour	Primary target for biopsy and treatment
Satellite Lesions	Secondary foci (prostate cancer is often multifocal)	May affect treatment strategy
Extracapsular Extension (ECE)	Tumour breaches the prostatic capsule	Staging: T3a — affects surgical planning

Lesion Type	Description	Clinical Significance
Seminal Vesicle Invasion (SVI)	Tumour extends into seminal vesicles	Staging: T3b — impacts prognosis

1.5.4 4.4 Organs at Risk (OARs) for Radiotherapy

Structure	Abbreviation	Why Segment?
Neurovascular Bundles	NVB	Nerve-sparing surgery — preserve potency
Rectal Wall	Rectum_Wall	Monitor tumour–rectum distance
Bladder Neck	Bladder_Neck	Preserve urinary continence

1.5.5 4.5 Segmentation Class Labels for the AI Model

Segment Name	Class Label	Modality	Clinical Goal
Whole Gland	`Prostate_WG`	T2W	Volume / PSA density calculation
Peripheral Zone	`PZ`	T2W	Cancer surveillance background
Transition Zone	`TZ`	T2W	BPH assessment background
Suspicious Lesion	`Lesion_PIRADS_{3|4|5}`	DWI/ADC	Targeted biopsy (MR-US fusion)
Seminal Vesicles	`SV`	T2W	Local staging (T3b)
Neurovascular Bundles	`NVB`	T2W	Nerve-sparing planning
Rectal Wall	`OAR_Rectum`	T2W	Radiotherapy constraints

1.5.6 4.6 SAM-Med3D-turbo Prompting Strategy for Prostate mpMRI

Context-dependent prompting: The same model must switch behaviour based on *where* the user clicks and *which sequence* is active.

1.5.6.1 Scenario 1: Anatomical Zone Segmentation (T2W)

```
1. User clicks bright outer rim on T2W axial
   → Model returns PZ mask
   → Expected Dice: ~0.90 (large, high-contrast structure)
```

```
2. User clicks central heterogeneous region on T2W
   → Model returns TZ mask

3. If mask leaks into rectum → place negative point on rectum
```

1.5.6.2 Scenario 2: Lesion Segmentation (DWI/ADC)

```
1. User clicks hypointense spot on ADC map
   → Model returns lesion mask (PI-RADS ≥4 region)

2. Use prior PZ mask as dense prompt context
   → Constrains lesion to within prostate boundary

3. Measure lesion volume → maps to PI-RADS size criterion
```

> **Critical note:** SAM-Med3D was pre-trained on **MR data** (SA-Med3D-140K includes MR modalities). However, the dataset card does not specify which MR sequences or whether prostate mpMRI is represented. Zero-shot performance on prostate zones may be moderate; fine-tuning on Charité prostate data will likely be necessary. Whole-gland segmentation (a large, well-defined structure) should work well zero-shot; zonal segmentation (PZ vs. TZ) is harder due to subtle signal differences.

1.5.7 4.7 Prostate vs. Coronary: Difficulty Comparison

Factor	Prostate	Coronary
Structure size	20–80 mL (large)	1.5–4 mm diameter (tiny)
Motion	None (static pelvis)	Cardiac motion (60–80 bpm)
Contrast	Good (gland vs. fat)	Variable (plaque vs. lumen)
Modality	MRI (multi-sequence)	CT (single phase)
Topology	Compact, roughly ellipsoidal	Thin, tortuous, branching tubes
Expected Dice	**> 0.90 (whole gland)**	**0.75–0.85 (lumen)**

1.6 5. Technical Challenges & Model-Aware Solutions

1.6.1 Challenge 1: Coronary Artery Motion Artefacts

Problem: Residual cardiac motion blurs vessel edges despite ECG gating. RCA most affected.

Solution:

```python
def motion_robust_segmentation(volume_4d, heart_rate):
    if heart_rate > 70:
        # Multi-phase reconstruction
        phases = extract_cardiac_phases(volume_4d, num_phases=10)
        masks = [model.segment(phase, points=prompt) for phase in phases]
        # Temporal median filter removes motion ghosts
        return np.median(masks, axis=0) > 0.5
    else:
        return model.segment(volume_4d[:,:,:,0], points=prompt)
```

Additional strategies: - Edge-preserving denoising (bilateral filter) pre-processing - Multi-point prompts every 5–10 mm along vessel to guide through corrupted regions - Auto-flag cases with edge sharpness < threshold for expert review

1.6.2 Challenge 2: Low-Attenuation Plaque Detection

Problem: Lipid-rich plaque (20–50 HU) nearly invisible against myocardium (50–70 HU).

Solution: Multi-stage approach: 1. Segment lumen and outer wall first (high-contrast boundaries) 2. Apply HU thresholding *within* the vessel wall mask 3. Use SAM-Med3D's ViT features for texture-based refinement (fine-tuning required)

1.6.3 Challenge 3: Dual-Wall Segmentation

Problem: Must segment both lumen and outer wall; wall thickness only 0.5–3 mm (1–5 voxels).

Solution: Sequential prompting (see §3.4B above).

1.6.4 Challenge 4: Prostate Zone Boundaries

Problem: PZ-TZ boundary is a gradual signal transition, not a sharp edge.

Solution: - Train multi-class model on annotated Charité prostate data - Use morphological priors (PZ wraps around TZ inferolaterally) - Negative prompts at zone transitions to sharpen boundaries

1.6.5 Challenge 5: Model Limitations — Honest Assessment

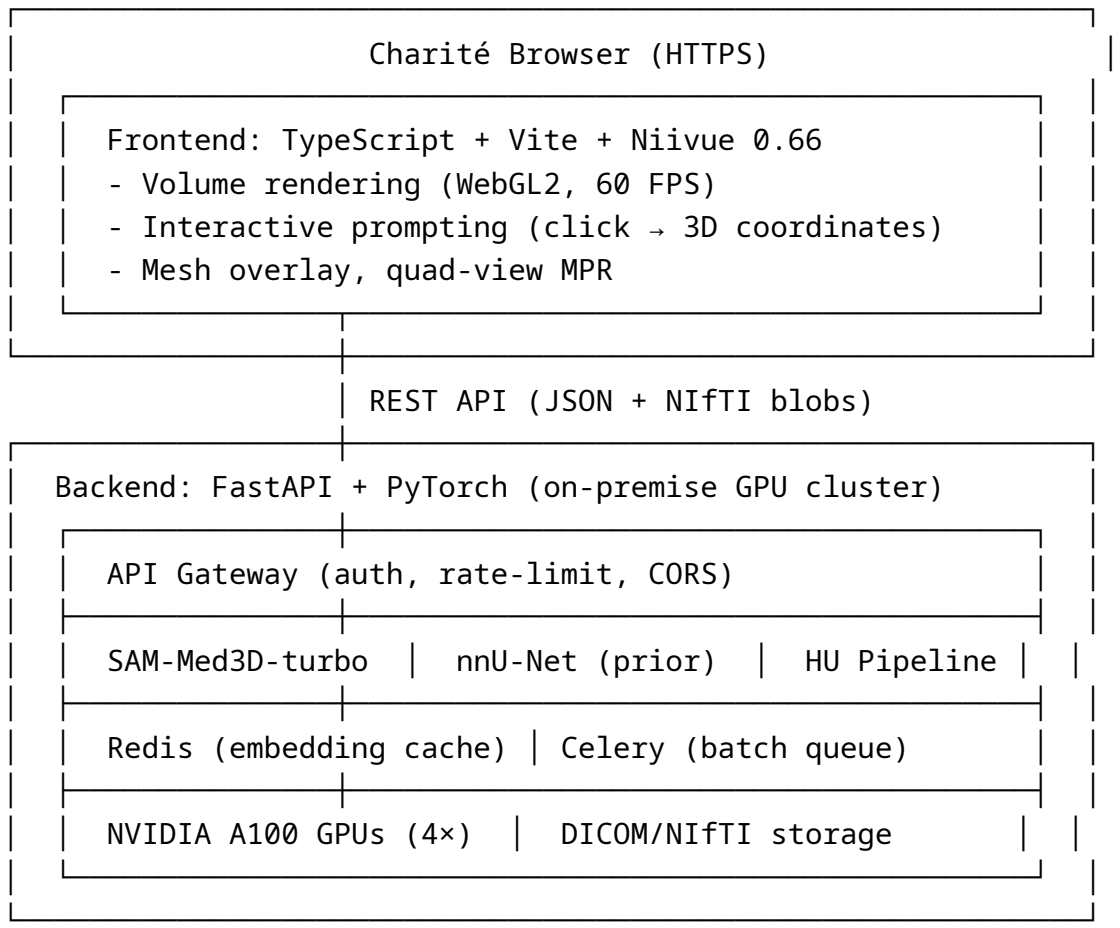
Limitation	Impact	Mitigation
No coronary CTA in pre-training	Zero-shot may underperform	Fine-tune on DISCHARGE annotations
Binary mask output only	Cannot directly predict PI-RADS score	Post-processing pipeline (volume, ADC stats)

Limitation	Impact	Mitigation
No `dense_prompt` in codebase	Lumen-as-prior strategy needs engineering	Custom prompt encoder modification
128^3 patch size constraint	Coronary arteries span > 128 voxels	Sliding window with overlap + stitching
No multi-class output	One mask per inference call	Multiple sequential inferences per case

---

1.7 6. System Architecture

1.7.1 6.1 High-Level Overview



1.7.2 6.2 Frontend Directory Structure

```
flow-segment-frontend/
├── src/
│   ├── main.ts                    # Entry point
```

```
│   ├── App.ts                    # Main app component
│   ├── components/
│   │   ├── NiivueViewer.ts      # Niivue canvas wrapper (single/quad)
│   │   ├── Toolbar.ts           # Top toolbar
│   │   ├── SegmentPanel.ts      # AI segmentation controls
│   │   ├── ResultsPanel.ts      # Plaque analysis / zone selector
│   │   ├── PromptHistory.ts     # Point prompt log + undo
│   │   └── MPRView.ts           # Multi-planar reconstruction
│   ├── services/
│   │   ├── api.ts               # Backend API client
│   │   ├── niivue.ts            # Niivue init & config
│   │   ├── loader.ts            # NIfTI.gz + DICOM loading
│   │   ├── medisParser.ts       # MEDIS TXT parsing
│   │   ├── medisMeshDirect.ts   # Client-side contour→mesh
│   │   ├── straightenedMPR.ts   # Frenet-Serret CPR
│   │   └── auth.ts              # LDAP/SSO
│   ├── types/
│   │   ├── volume.ts
│   │   ├── segmentation.ts
│   │   ├── mesh.ts
│   │   └── api.ts
│   └── utils/
│       ├── coordinates.ts       # 3D coordinate transforms
│       ├── meshGenerator.ts     # Marching cubes (vtk.js)
│       └── export.ts            # NIfTI / DICOM-SEG / CSV export
├── package.json
├── tsconfig.json
├── vite.config.ts
└── index.html
```

1.7.3 6.3 Backend Directory Structure

```
flow-segment-backend/
├── app/
│   ├── main.py                  # FastAPI application
│   ├── config.py                # Environment config
│   ├── models/
│   │   ├── sam_med3d.py         # SAM-Med3D-turbo wrapper
│   │   ├── nnu_net.py           # nnU-Net (anatomical prior)
│   │   └── plaque_analyser.py   # HU-based plaque classification
│   ├── api/
│   │   ├── segment.py           # Segmentation endpoints
│   │   ├── volumes.py           # Volume management
```

```
│   │   ├── mesh.py                # Mesh generation (nii2mesh)
│   │   └── auth.py                # LDAP/SSO
│   └── services/
│       ├── cache.py               # Redis embedding cache
│       ├── dicom_processor.py     # DICOM → NIfTI (SimpleITK)
│       ├── mesh_generator.py      # nii2mesh wrapper
│       └── registration.py        # Elastix (longitudinal)
├── Dockerfile
├── docker-compose.yml
├── requirements.txt
└── README.md
```

1.8 7. Clinical Workflows

1.8.1 7.1 Workflow 1: Interactive Coronary Segmentation (Radiologist)

1. Radiologist opens browser logs in (Charité SSO)
2. Loads DISCHARGE CCTA scan from PACS
3. Clicks proximal LAD + distal tip AI segments entire lumen in < 2 s
4. Optionally: places negative point to prune vein leakage
5. Clicks “Expand to Wall” second inference outer wall mask
6. Right panel shows: stenosis %, plaque composition, volume
7. Exports segmentation (NIfTI / DICOM-SEG / CSV report)

1.8.2 7.2 Workflow 2: Batch Processing for DISCHARGE Research

1. Research coordinator uploads 100 DISCHARGE cases
2. AI processes overnight (Celery + multi-GPU batch mode)
3. Quality control: auto-flag cases with Dice < 0.75 or disconnected components
4. Expert reviews flagged cases corrections feed active-learning loop
5. Export refined segmentations for MACE prediction analysis

1.8.3 7.3 Workflow 3: MEDIS TXT + Mesh + Straightened MPR (Legacy)

1. Load CTA volume (NIfTI.gz) in browser
2. Load MEDIS TXT file (expert contour rings)
3. Client-side mesh generation: parse TXT NVMesh (50–100 ms)
4. Overlay lumen + vessel wall meshes on CTA
5. Generate straightened MPR for longitudinal plaque assessment

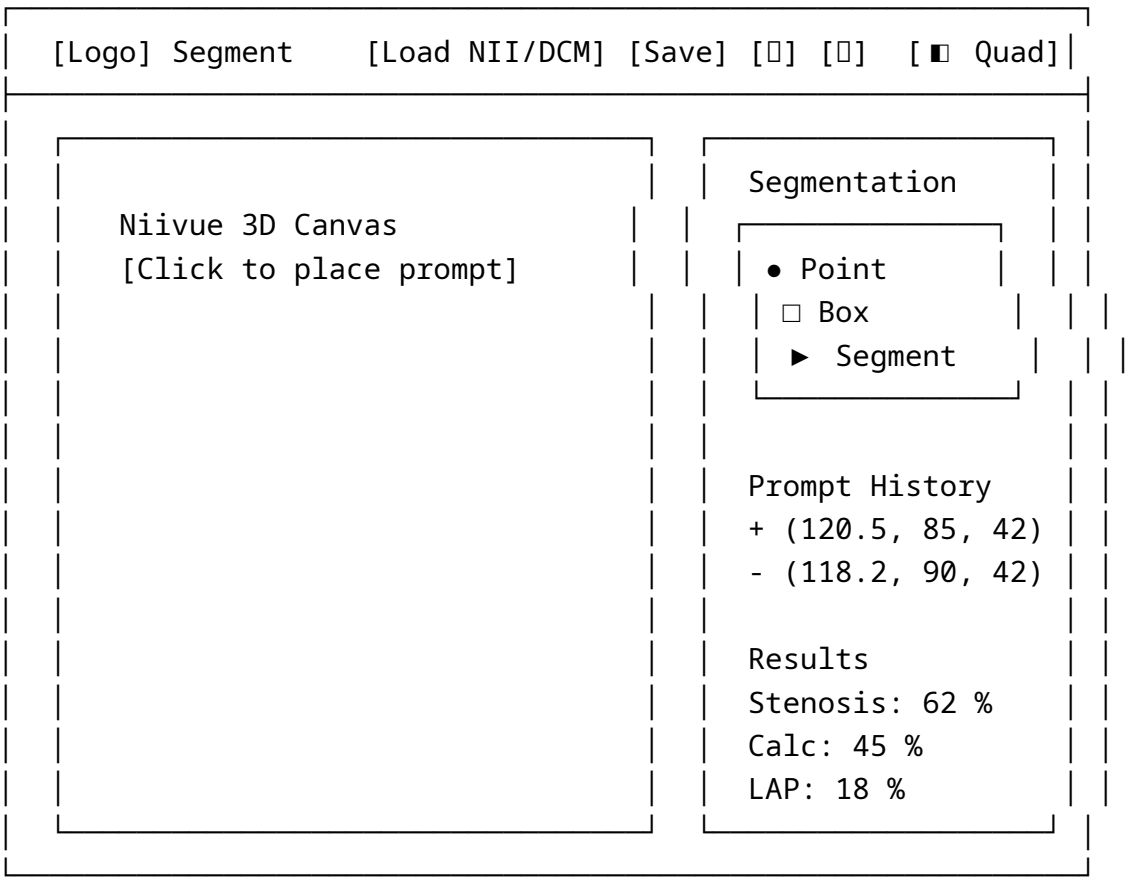
---

1.9 8. Frontend: Niivue Viewer & UI/UX

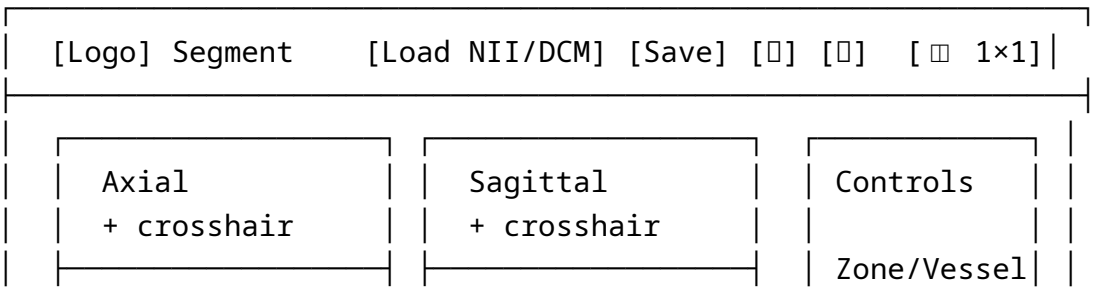
1.9.1 8.1 Key Niivue v0.66.0 Capabilities

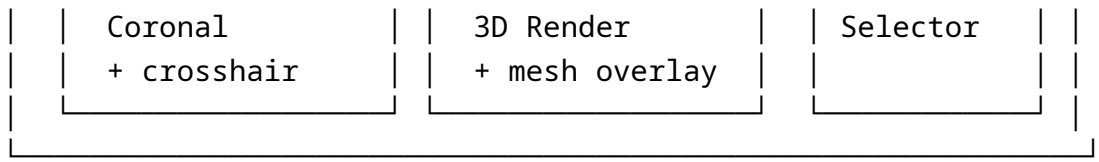
- □ WebGL2 rendering: 60 FPS for 512^3 volumes
- □ `onLocationChange` / `onMouseUp` □ capture 3D mm coordinates for prompts
- □ Multi-planar reconstruction (axial / coronal / sagittal sync)
- □ `nv.addMesh()` for 3D surface overlay with adjustable opacity
- □ Full TypeScript definitions
- □ In-browser DICOM via dcm2niix-wasm

1.9.2 8.2 UI Layout: Single View (Default)



1.9.3 8.3 UI Layout: Quad View (MPR + 3D)





1.9.4 8.4 Design Principles

- **Dark theme** (#1a1a1a) — reduces eye strain for long sessions
- **Minimal chrome** — maximise canvas area
- **Radiologist-first** — optimised for clinical workflow
- Primary: Blue (#3b82f6), Accent: Green (#10b981), Warning: Orange (#f59e0b), Error: Red (#ef4444)

1.10 9. Backend: Inference Pipeline & API

1.10.1 9.1 API Endpoints

```
# Segmentation (AI)
POST /api/segment/point        # Point-based prompting (SAM-Med3D)
POST /api/segment/box          # Bounding box prompting
POST /api/segment/refine       # Refine with negative points

# Volume Management
POST /api/volumes/upload       # Upload DICOM / NIfTI
GET  /api/volumes/{id}         # Retrieve processed volume

# Mesh Generation
POST /api/mesh/from-mask       # Mask → MZ3 mesh (nii2mesh)
POST /api/mesh/quick           # Fast preview (marching cubes)

# MEDIS TXT Processing
POST /api/medis/upload         # Upload MEDIS TXT
POST /api/medis/to-mesh        # Convert contours to mesh

# Straightened MPR
POST /api/straighten/create    # Centerline → straightened volume

# Batch Processing
POST /api/batch/process        # Queue batch of cases
GET  /api/batch/{job_id}       # Job status

# Authentication
```

```
POST /api/auth/login           # LDAP / Charité SSO
GET  /api/health               # Health check + GPU status
```

1.10.2 9.2 Segmentation Endpoint (Core)

```python
from fastapi import APIRouter
import nibabel as nib
import numpy as np
import medim

router = APIRouter()

# Model loaded once at startup
model = medim.create_model(
    "SAM-Med3D", pretrained=True,
    checkpoint_path="app/models/checkpoints/sam_med3d_turbo.pth"
)
model = model.cuda().half().eval()

@router.post("/api/segment/point")
async def segment_point(
    volume_id: str,
    coordinates: list[float],   # [x, y, z] in mm
    labels: list[int] = [1],    # 1=positive, 0=negative
):
    # Load volume from cache/disk
    vol_nii = nib.load(f"/data/volumes/{volume_id}.nii.gz")
    volume = vol_nii.get_fdata()

    # Convert mm → voxel coordinates
    voxel_coords = np.linalg.inv(vol_nii.affine) @ [*coordinates, 1]

    # Run SAM-Med3D inference
    mask = model.segment(volume, points=[voxel_coords[:3]], labels=labels)

    # Return mask as NIfTI
    mask_nii = nib.Nifti1Image(mask.astype(np.uint8), vol_nii.affine)
    # ... serialize and return
```

1.11 10. MEDIS TXT Legacy Pipeline

1.11.1 10.1 Format Description

MEDIS TXT contains vessel wall contours from MEDIS QAngio CT: - **Lumen** contours: define inner wall (blood pool boundary) - **VesselWall** contours: define outer wall (including plaque) - Each contour: group label, slice distance, N points in 3D mm coordinates

1.11.2 10.2 Parser (TypeScript)

```
export interface MedisContour {
  group: "Lumen" | "VesselWall";
  sliceDistance: number;           // mm along vessel
  points: { x: number; y: number; z: number }[];
}

export function parseMedisTxt(content: string): MedisContour[] {
  const lines = content.split("\n");
  const contours: MedisContour[] = [];
  let current: Partial<MedisContour> = {};
  let points: { x: number; y: number; z: number }[] = [];

  for (const line of lines) {
    if (line.startsWith("# group:")) {
      current.group = line.split(":")[1].trim() as "Lumen" | "VesselWall";
    } else if (line.startsWith("# SliceDistance:")) {
      current.sliceDistance = parseFloat(line.split(":")[1]);
    } else if (line.startsWith("# Contour index:")) {
      if (current.group && points.length > 0) {
        contours.push({ ...current, points } as MedisContour);
      }
      points = [];
    } else if (line.trim() && !line.startsWith("#")) {
      const [x, y, z] = line.trim().split(/\s+/).map(Number);
      if (!isNaN(x)) points.push({ x, y, z });
    }
  }
  if (current.group && points.length > 0) {
    contours.push({ ...current, points } as MedisContour);
  }
  return contours;
}
```

1.11.3 10.3 Direct Client-Side Mesh Construction (< 100 ms)

Contour rings are connected into a tube mesh directly in the browser — no backend round-trip needed. Algorithm: connect ring N to ring N+1 with triangle pairs.

Performance comparison: | Method | Latency | Network | |——–|———|———| | Backend (buildstl.py STL download) | 500–2000 ms | Required | | **Client-side (TXT NVMesh)** | **50–100 ms** | **None** |

1.12 11. Mesh Generation Strategies

1.12.1 11.1 Three Approaches

Approach	Method	Speed	Quality	Use Case
Ultra-Simple	Connect contour rings STL	< 50 ms	Faceted	MEDIS export
Client-side vtk.js	Marching cubes on mask	< 1 s	Good	Interactive preview
Server nii2mesh MZ3	Decimation + smoothing	1–3 s	High	Final visualisation

1.12.2 11.2 Recommended Web Format: MZ3

- 3–5× smaller than PLY, 10× smaller than STL
- Binary, gzip-compressed, native Niivue support
- Target: < 5 MB per mesh, 50K–200K triangles, 60 FPS rendering

---

1.13 12. Centerline Extraction & Straightened MPR (CPR)

1.13.1 12.1 Overview

Straightened MPR (Curved Planar Reformation) “unfolds” a tortuous vessel into a straight view. Essential for assessing stenosis and plaque distribution along the entire vessel length.

Three steps: 1. **Centerline extraction** — centroid of lumen contours (from MEDIS) or Voronoi skeletonisation (from AI mask) 2. **Frenet-Serret frame** — compute Tangent (T), Normal (N), Binormal (B) at each point 3. **Cross-section sampling** — extract perpendicular slices stack into straightened volume

1.13.2 12.2 Mathematical Foundation

Tangent (finite differences):

```
T[i] = normalize(centerline[i+1] - centerline[i-1]) // central difference
```

Normal (curvature direction):

```
dT = T[i+1] - T[i-1]
N[i] = normalize(dT)  // or arbitrary perpendicular if straight
```

Binormal: `B[i] = T[i] × N[i]`

Cross-section point: `Q(u,v) = P[i] + u·N[i] + v·B[i]`

1.13.3 12.3 Interactive Controls

- **Position slider:** Select centerline point (0 □ N-1)
- **Rotation slider:** Rotate cross-section around T (Rodrigues' formula)
- **Zoom slider:** Adjust cross-section FOV
- **Quad-view:** CTA overview | cross-section | straightened MPR | 3D mesh

---

1.14 13. General-Purpose Rotatable Volume Viewer

Port of legacy `viewer.py` (PyQtGraph):

- Free rotation via mouse drag (Rodrigues' rotation formula)
- Three-panel orthogonal views (YZ, XZ, XY) — all rotate together
- Drag modes: Rotation (0), Paint/Label (1), Window/Level (2)
- Real-time volume resampling (SimpleITK □ WebGL equivalent)
- Target: 30–60 FPS during rotation

1.15 14. Deployment, Performance & Security

1.15.1 14.1 Infrastructure

Component	Technology
Containerisation	Docker + NVIDIA Container Toolkit
GPU	4× NVIDIA A100 (80 GB each)
Embedding cache	Redis (sub-second for repeated prompts)
Batch queue	Celery + Redis broker
Authentication	LDAP / Charité SSO
Compliance	GDPR (all data on-premise, no cloud)

1.15.2 14.2 Performance Targets

Metric	Target
Single-click → mask	< 2 s end-to-end
Embedding computation (first prompt)	~1 s
Subsequent prompts (cached embedding)	< 0.5 s
Mesh generation (MZ3)	< 3 s
Batch throughput	~50 cases / hour (4 GPUs)
CTA volume memory	~200 MB ($512^3 \times 2$ bytes)

1.15.3 14.3 Memory Budget

Component	Size
CTA volume (512^3)	~200 MB
SAM-Med3D-turbo (FP16)	~4 GB VRAM
Embedding cache (per volume)	~500 MB
Straightened volume ($64^2 \times 200$)	~8 MB
Mesh (MZ3, per vessel)	< 5 MB

1.16 15. Research Roadmap & Milestones

1.16.1 Phase 1: MVP — MEDIS TXT Visualisation (Weeks 1–4)

- ☐ MEDIS TXT parser + client-side mesh in Niivue
- ☐ Centerline extraction + straightened MPR
- ☐ Quad-view layout with interactive sliders
- ☐ NIfTI.gz / DICOM loading

1.16.2 Phase 2: SAM-Med3D Integration (Weeks 5–8)

- ☐ Backend: load turbo checkpoint, expose `/api/segment/point`
- ☐ Frontend: click → prompt → mask overlay
- ☐ Redis embedding cache for sub-second repeated prompts
- ☐ Dual-wall sequential segmentation pipeline

1.16.3 Phase 3: DISCHARGE Evaluation (Weeks 9–12)

- ☐ Zero-shot baseline on held-out DISCHARGE cases
- ☐ Quantify Dice, Hausdorff, stenosis % correlation vs. expert
- ☐ Identify failure modes (motion, calcification, bifurcations)

1.16.4 Phase 4: Fine-tuning & Active Learning (Weeks 13–20)

- ☐ Fine-tune SAM-Med3D on DISCHARGE annotations (nnU-Net-style data prep)
- ☐ Active-learning loop: expert corrections weekly re-training
- ☐ Benchmark against task-specific nnU-Net baseline

1.16.5 Phase 5: Prostate Extension (Weeks 21–28)

- ☐ Adapt pipeline for prostate mpMRI (multi-sequence input)
- ☐ Zone-specific class labels (PZ / TZ / lesion)
- ☐ Validate on Charité prostate cohort

1.16.6 Phase 6: Clinical Validation & Publication (Weeks 29–36)

- ☐ Prospective reader study (Dice vs. time vs. inter-observer)
- ☐ Open-source release + MedSegFM competition baseline
- ☐ Publication: “Foundation-model-assisted coronary CTA segmentation at scale”

1.16.7 Quarterly Research Milestones

Quarter	Milestone
Q1 2026	Zero-shot baseline on DISCHARGE + MVP deployed
Q2 2026	Active-learning loop running; Dice 0.80 on coronary lumen
Q3 2026	Prospective reader study; prostate pipeline validated
Q4 2026	Open-source release; conference/journal submission

1.17 16. References & Resources

1.17.1 16.1 Core References

1. **Wang H, Guo S, Ye J, Deng Z, Cheng J, Li T, Chen J, Su Y, Huang Z, Shen Y, Fu B, Zhang S, He J, Qiao Y.** SAM-Med3D: Towards General-purpose Segmentation Models for Volumetric Medical Images. *ECCV BIC 2024 (Oral)*. arXiv:2310.15161. Paper · GitHub · Checkpoint · Dataset
2. **Dewey M et al.** Cardiac CT for the diagnosis of coronary artery disease in patients with stable chest pain (DISCHARGE). *NEJM 2022*.

3. **Isensee F, Jaeger PF, Kohl SAA, Petersen J, Maier-Hein KH.** nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods 2021; 18: 203–211*.
4. **Niivue** — WebGL2 medical image viewer. GitHub v0.66.0
5. **Kirillov A et al.** Segment Anything. *ICCV 2023*. (SAM — the 2D foundation)

1.17.2 16.2 Related Medical Segmentation Models

Model	Dimension	Modalities	Prompts	Key Difference from SAM-Med3D
SAM (Meta)	2D	Natural images	Point/box/text	No medical training, 2D only
SAM-Med2D	2D	Medical (slice-wise)	Point/box	2D □ cannot capture volumetric context
MedSAM	2D	Medical (slice-wise)	Box only	Simpler architecture, box prompts only
SAM-Med3D	**3D**	**Medical (volumetric)**	**3D point**	**Native 3D — our choice**
nnU-Net	3D	Medical (task-specific)	None (automatic)	Not promptable; requires per-task training

1.17.3 16.3 Official Citation

```
@misc{wang2024sammed3dgeneralpurposesegmentationmodels,
  title     = {SAM-Med3D: Towards General-purpose Segmentation Models
               for Volumetric Medical Images},
  author    = {Haoyu Wang and Sizheng Guo and Jin Ye and Zhongying Deng
               and Junlong Cheng and Tianbin Li and Jianpin Chen
               and Yanzhou Su and Ziyan Huang and Yiqing Shen
               and Bin Fu and Shaoting Zhang and Junjun He and Yu Qiao},
  year      = {2024},
  eprint    = {2310.15161},
  archivePrefix = {arXiv},
  primaryClass  = {cs.CV},
  url       = {https://arxiv.org/abs/2310.15161},
}
```

This document is the living foundation of the Charité Segment Platform research project. All claims about SAM-Med3D are verified against the published paper (arXiv:2310.15161), official

GitHub README, and HuggingFace model/dataset cards. Critical caveats about zero-shot performance on coronary CTA and prostate mpMRI (neither directly evaluated in the paper) are noted throughout.