

Technische Dokumentation: Immobilien-Transaktionsplattform

Version: 1.0 Datum: 22.12.2025 Datei: `streamlit_app.py` (~36.600 Zeilen)

Inhaltsverzeichnis

- Übersicht & Architektur
- Benutzerrollen & Dashboards
- Workflow-Engine
- Datenmodelle
- Hilfsfunktionen
- Kommunikationsschnittstellen
- Bekannte Einschränkungen

1. Übersicht & Architektur

1.1 Technologie-Stack

Komponente	Technologie
Frontend	Streamlit (Python)
Datenspeicherung	Session-State (RAM)
Styling	Custom CSS (injiziert)
PDF-Verarbeitung	ReportLab, PyPDF2
OCR	Claude Vision API / Tesseract

1.2 Hauptmodule

```
streamlit_app.py
├─ Datenklassen (Z. 1837-6800)
│  └─ Enums (UserRole, ProjektStatus, etc.)
│  └─ Dataclasses (User, Projekt, Akte, etc.)
│  └─ Workflow-Template (WORKFLOW_TEMPLATE_KV)
├─ Hilfsfunktionen (Z. 6800-9500)
│  └─ Session-Management
│  └─ Benachrichtigungen
│  └─ Berechnungen (GNotKG, Kosten)
│  └─ Validierungen
├─ Dashboard-Views (Z. 13000-32000)
│  └─ Käufer-Dashboard (Z. 14575)
│  └─ Verkäufer-Dashboard (Z. 18163)
│  └─ Makler-Dashboard (Z. 13336)
│  └─ Notar-Dashboard (Z. 21289)
├─ Rendering-Funktionen (Z. 32000-36000)
│  └─ Menü-Rendering
│  └─ Formulare
│  └─ Styling
├─ Main-App (Z. 36000+)
│  └─ Login/Registrierung
│  └─ Routing
│  └─ Session-Initialisierung
```

2. Benutzerrollen & Dashboards

2.1 Rollenübersicht

Rolle	Enum-Wert	Dashboard-Funktion	Zeile
Käufer	KAEUFER	kaeufer_dashboard()	14575
Verkäufer	VERKAEUFER	verkaeufer_dashboard()	18163
Makler	MAKLER	makler_dashboard()	13336
Notar	NOTAR	notar_dashboard()	21289
Finanzierer	FINANZIERER	finanzierer_dashboard()	27908
Notar-Mitarbeiter	NOTAR_MITARBEITER	notarmitarbeiter_dashboard()	30495

Rolle	Enum-Wert	Dashboard-Funktion	Zeile
Makler-Mitarbeiter	MAKLER_MITARBEITER	maklermitarbeiter_dashboard()	31152

2.2 Käufer-Dashboard (14 Tabs)

Tab	Funktion	Beschreibung
Mein Portal	render_mandanten_portal()	Übersicht und Schnellzugriff
Timeline	kaeufer_timeline_view()	Projekt-Fortschritt
Projekte	kaeufer_projekte_view()	Exposé, Preisverhandlung
Aufgaben	kaeufer_aufgaben_view()	Todo-Liste mit Kategorien
Finanzierung	kaeufer_finanzierung_view()	6 Sub-Tabs für Kredit
Handwerker	kaeufer_handwerker_empfehlungen()	Notar-Empfehlungen
Ausweis	OCR-Erfassung	Personalausweis-Upload
Nachrichten	kaeufer_nachrichten()	Kommunikation
Dokumente	kaeufer_dokumente_view()	Dokumenten-Download
Vertragsvergleich	render_vertragsvergleich_tab()	Side-by-Side Diff
Termine	render_termin_kalender()	Kalender-Ansicht
Papierkorb	render_papierkorb_tab()	Gelöschte Elemente
Vorlesen	render_tts_einstellungen()	Text-to-Speech

Besondere Features: - Finanzierungsrechner mit bis zu 4 Modell-Vergleichen -
Kaufnebenkosten-Rechner (bundeslandabhängig) - Ideenboard für Renovierungsideen -
System-generierte Aufgaben basierend auf Projektstatus

2.3 Verkäufer-Dashboard (14 Tabs)

Tab	Funktion	Beschreibung
Mein Portal	Übersicht	Dashboard-Startseite
Timeline	verkaeufer_timeline_view()	Fortschritt
Projekte	verkaeufer_projekte_view()	Projektübersicht

Tab	Funktion	Beschreibung
Preisfindung	<code>verkaeufer_preisfindung_view()</code>	Marktanalyse
Makler finden	<code>verkaeufer_makler_finden()</code>	Geprüfte Makler
Ausweis	OCR-Erfassung	Daten-Upload
Dokumente	<code>verkaeufer_dokumente_view()</code>	Upload & Freigaben
Anforderungen	<code>render_document_requests_view()</code>	Dokumentenanfragen
Nachrichten	<code>verkaeufer_nachrichten()</code>	Kommentare
Eigene Kosten	<code>verkaeufer_eigene_kosten_view()</code>	Kostenberechnung
Vertragsvergleich	Diff-Ansicht	Vertragsversionen
Termine	Kalender	Terminübersicht
Papierkorb	Gelöschte Elemente	Wiederherstellung
Vorlesen	TTS	Vorlesefunktion

2.4 Makler-Dashboard (19 Tabs)

Tab	Funktion	Beschreibung
Timeline	<code>makler_timeline_view()</code>	Workflow-Übersicht
Projekte	<code>makler_projekte_view()</code>	Projekt-Verwaltung
Marktanalyse	<code>makler_marktanalyse_view()</code>	Vergleichsobjekte
Profil	<code>makler_profil_view()</code>	Firmen-/Team-Daten
Rechtsdokumente	<code>makler_rechtliche_dokumente()</code>	AGB, Datenschutz
Teilnehmer-Status	<code>makler_teilnehmer_status()</code>	Akzeptanz-Tracking
Einladungen	<code>makler_einladungen()</code>	Käufer/Verkäufer einladen
Kommentare	<code>makler_kommentare()</code>	Nachrichten
Ausweis	<code>makler_ausweis_erfassung()</code>	Daten erfassen
Mitarbeiter	<code>render_makler_mitarbeiter_verwaltung()</code>	Team-Management

Tab	Funktion	Beschreibung
Kommunikation	<code>render_kommunikationszentrale()</code>	Postfach
Vertragsvergleich	Diff-Ansicht	-
Fristen	<code>render_fristenmanagement()</code>	Deadline-Tracking
Reporting	<code>render_reporting_dashboard()</code>	KPIs
Papierkorb	Gelöschte Elemente	-
Vorlesen	TTS	-
DSGVO	<code>render_dsgvo_tab_notar()</code>	Datenschutz

Berechtigungssystem für Mitarbeiter:

MaklerBerechtigungTyp:

- PROJEKTE_ANSEHEN, PROJEKTE_ERSTELLEN, PROJEKTE_BEARBEITEN
- NACHRICHTEN_SENDEN, NACHRICHTEN_LESEN, IM_NAMEN_KOMMUNIZIEREN
- DOKUMENTE_HOCHLADEN, EXPOSE_ERSTELLEN
- TEILNEHMER_EINLADEN, TERMINE_ERSTELLEN
- PREISE_SEHEN, PREISE_VERHANDELN
- MITARBEITER_VERWALTEN, EINSTELLUNGEN_AENDERN

2.5 Notar-Dashboard (5 Hauptgruppen)

NOTAR_MENU_STRUKTUR:

```

├─ Timeline (📅)
│   ├── Übersicht → notar_timeline_view()
│   └── Berichte → render_reporting_dashboard()
├─ Akte (📁)
│   ├── Projekte → notar_projekte_view()
│   ├── Verwaltung → notar_aktenmanagement_view()
│   └── Import → notar_aktenimport_view()
├─ Verträge (📝)
│   ├── Archiv → notar_vertragsarchiv_view()
│   ├── Erstellen → notar_vertragserstellung_view()
│   ├── Kaufvertrag → notar_kaufvertrag_generator()
│   ├── Vergleich → render_vertragsvergleich_tab()
│   └── Vorlagen → render_vorlagen_management()
├─ Termine (📅 17)
│   ├── Kalender → notar_termine()
│   ├── Fristen → render_fristenmanagement()
│   └── Checklisten → notar_checklisten_view()
├─ Mehr (☰)
│   └── Dokumente

```

```

|   |— Ermittlung → notar_datenermittlung_view()
|   |— Anforderung → render_document_requests_view()
|   |— Freigaben → notar_dokumenten_freigaben()
|   |— Rechtsdoku → notar_rechtsdokumente_view()
|— Finanzen
|   |— Preise → notar_preiseinigungen_view()
|   |— Finanzierung → notar_finanzierungsnachweise()
|— Kontakte
|   |— Mitarbeiter → notar_mitarbeiter_view()
|   |— Ausweise → notar_ausweis_erfassung()
|   |— Makler → notar_makler_empfehlung_view()
|   |— Handwerker → notar_handwerker_view()
|— Nachrichten → render_kommunikationszentrale()
|— System
|   |— Einstellungen → notar_einstellungen_view()
|   |— DSGVO → render_dsgvo_tab_notar()
|   |— Papierkorb → render_papierkorb_tab()
|   |— Vorlesen → render_tts_einstellungen()

```

3. Workflow-Engine

3.1 Konfiguration (WORKFLOW_TEMPLATE_KV)

```

WORKFLOW_TEMPLATE_KV = {
  "template_id": "WT_KV_MODERN_V1",
  "name": "Kaufvertrag Immobilien (Modern)",
  "version": "1.0.0",

  "segments": [
    {"segment_id": "O_ONBOARDING", "label": "Onboarding", "order": 0},
    {"segment_id": "A_PRE_BEURKUNDUNG", "label": "Vor Beurkundung", "order": 1},
    {"segment_id": "B_POST_BEURKUNDUNG_PRE_FAELLIGKEIT", "label": "Nach Beurkundung", "order": 2},
    {"segment_id": "C_POST_FAELLIGKEIT_PRE_UEBERGABE", "label": "Kaufpreisabwicklung", "order": 3},
    {"segment_id": "D_POST_UEBERGABE_PRE_EINTRAGUNG", "label": "Abschluss", "order": 4}
  ],

  "milestones": [
    {"milestone_type": "ONBOARDING_ABGESCHLOSSEN", "label": "Onboarding", "order": 0},
    {"milestone_type": "NOTARTERMIN_BEURKUNDUNG", "label": "Beurkundung", "order": 1},
    {"milestone_type": "KAUFPREISFAELLIGKEIT", "label": "Kaufpreisfälligkeit", "order": 2},
    {"milestone_type": "SCHLUESSELUEBERGABE", "label": "Schlüsselübergabe", "order": 3},
    {"milestone_type": "EIGENTUMSUMSCHREIBUNG", "label": "Eigentumsumschreibung", "order": 4}
  ],

  "progress_weights": {
    "ONBOARDING_ABGESCHLOSSEN": 0.20,

```

```

    "NOTARTERMIN_BEURKUNDUNG": 0.20,
    "KAUFPREISFAELLIGKEIT": 0.20,
    "SCHLUESSELUEBERGABE": 0.20,
    "EIGENTUMSUMSCHREIBUNG": 0.20
  }
}

```

3.2 Workflow-Steps

Segment O: Onboarding (7 Steps)

Code	Titel	Verantwortlich	Dependencies
O_ERSTKONTAKT	Erstkontakt/ Anfrage	Makler	-
O_OBJEKTDATEN_ERFASST	Objektdaten erfasst	Makler	O_ERSTKONTAKT
O_KAUFANGEBOT_ANGENOMMEN	Kaufangebot angenommen	Makler	O_OBJEKTDATEN_ERFASST
O_FINANZIERUNG_BESTAETIGT	Finanzierung bestätigt	Käufer	O_KAUFANGEBOT (bedingt)
O_KAEUFER_DATEN_VOLLSTAENDIG	Käufer-Daten vollständig	Käufer	O_KAUFANGEBOT
O_VERKAEUFER_DATEN_VOLLSTAENDIG	Verkäufer- Daten vollständig	Verkäufer	O_KAUFANGEBOT
O_NOTARAUFTRAG_ERTEILT	Notarauftrag erteilt	Makler	O_KAEUFER + O_VERKAEUFER

Segment A: Vor Beurkundung (10 Steps)

Code	Titel	Dependencies
A_AKTE_ANGELEGT	Akte angelegt	O_NOTARAUFTRAG_ERTEILT
A_PARTEIEN_ERFASST	Parteien erfasst	A_AKTE_ANGELEGT
A_AUSWEISE_VOLLSTAENDIG	Ausweise vollständig	A_PARTEIEN_ERFASST
A_GRUNDBUCH_IMPORT_GEPRUEFT	Grundbuch geprüft	A_AKTE_ANGELEGT
A_ENTWURF_ERSTELLT	Vertragsentwurf erstellt	

Code	Titel	Dependencies
		A_GRUNDBUCH + A_PARTEIEN
A_DOKUMENTE_ANGEFORDERT	Dokumente angefordert	A_ENTWURF_ERSTELLT
A_FREIGABEN_EINGEHOLT	Freigaben eingeholt	A_ENTWURF_ERSTELLT
A_TERMIN_BESTAETIGT	Notartermin bestätigt	A_FREIGABEN_EINGEHOLT
A_BEURKUNDUNG_DOKUMENTIERT	Beurkundung dokumentiert	A_TERMIN + A_AUSWEISE + ...

Segment B: Nach Beurkundung (8 Steps) - B_VOLLZUG_GESTARTET →

B_AUFLASSUNGSVORMERKUNG_EINGETRAGEN - B_VORKAUFSRECHT_ANGEFragt
→ B_VORKAUFSRECHT_BESCHIEDEN - B_GRUNDSCHULD_* (3 bedingte Steps bei
Finanzierung) - B_FAELLIGKEITSMITTEILUNG_VERSANDT

Segment C: Kaufpreisabwicklung (5 Steps) - C_KAUFPREIS_EINGEGANGEN →

C_UNBEDENKLICHKEIT_EINGEGANGEN - C_LOESCHUNGSBEWILLIGUNG_ERHALTEN →
C_SCHLUESSELUEBERGABE_DOKUMENTIERT

Segment D: Abschluss (3 Steps) - D_AUFLASSUNG_BEANTRAGT →

D_EINTRAGUNG_BESTAETIGT → D_ABSCHLUSS_ARCHIV

3.3 Workflow-Funktionen

```
# Steps für Segment abrufen
get_workflow_steps_for_segment(segment_id, include_conditional, financing_required)

# Dependencies eines Steps
get_step_dependencies(step_code) → List[str]

# Status berechnen (DONE, OPEN, BLOCKED, SKIPPED)
calculate_step_status(step_code, completed_steps, financing_required)

# Gesamtfortschritt
calculate_workflow_progress(completed_steps, financing_required) → {
    "total_progress": float, # 0-100%
    "segments": {...},      # pro Segment: done/total
    "milestones": {...}     # pro Meilenstein: done/label
}
```


4. Datenmodelle

4.1 Benutzer & Authentifizierung

```
@dataclass
class User:
    user_id: str
    name: str
    email: str
    rolle: str # UserRole Enum
    password_hash: str
    projekt_ids: List[str]
    onboarding_complete: bool
    document_acceptances: List[DocumentAcceptance]
    notifications: List[str]
    personal_daten: Optional[PersonalDaten]
    ausweis_foto: Optional[bytes]

@dataclass
class PersonalDaten:
    vorname, nachname, geburtsname: str
    geburtsdatum: Optional[date]
    geburtsort, nationalitaet: str
    strasse, hausnummer, plz, ort: str
    ausweisnummer, ausweisart: str
    ausstellungsbehoerde: str
    ausstellungsdatum, gueltig_bis: Optional[date]
    ocr_vertrauenswuerdigkeit: float # 0.0-1.0
    manuell_bestatigt: bool
```

4.2 Projekte & Immobilien

```
@dataclass
class Projekt:
    projekt_id: str
    name: str
    beschreibung: str
    adresse: str
    kaufpreis: float
    expose_pdf: Optional[bytes]

    # Beteiligte
    makler_id: str
    kaeufer_ids: List[str]
    verkaeufer_ids: List[str]
    finanzierer_ids: List[str]
    notar_id: str
```

```
# Status & Workflow
status: str # ProjektStatus Enum
workflow_completed_steps: List[str]
financing_required: bool

# Meilensteine
milestone_beurkundung_planned: Optional[datetime]
milestone_beurkundung_actual: Optional[datetime]
# ... weitere Meilenstein-Felder

@dataclass
class ExposeData:
    expose_id: str
    projekt_id: str
    objekttitel: str
    objektart: str # Wohnung, Haus, etc.

    # Flächen & Räume
    wohnflaeche: float
    grundstuecksflaeche: float
    anzahl_zimmer: float

    # Ausstattung (bool-Felder)
    hat_balkon, hat_terrasse, hat_garten: bool
    hat_garage, hat_stellplatz: bool
    hat_fahrstuhl, hat_sauna: bool

    # Preise
    kaufpreis: float
    preis_pro_qm: float
    hausgeld: float
    grundsteuer: float

    # Energieausweis
    energieeffizienzklasse: str
    endenergieverbrauch: float

    # Grundbuch
    gemarkung, flur, flurstueck: str
    grundbuchamt, grundbuchblatt: str

    # Bilder
    titelbild: Optional[bytes]
    weitere_bilder: List[bytes]
    grundrisse: List[bytes]
```

4.3 Dokumente

```
@dataclass
class VerkäuferDokument:
    dokument_id: str
    verkaeufer_id: str
    projekt_id: str
    dokument_typ: str # DokumentTyp Enum
    dateiname: str
    pdf_data: bytes

    # Freigaben
    freigegeben_fuer_makler: bool
    freigegeben_fuer_notar: bool
    freigegeben_fuer_finanzierer: bool
    freigegeben_fuer_kaeufer: bool

    status: str
    upload_datum: datetime

# DokumentTyp Enum (40+ Typen):
PERSONAL AUSWEIS, REISEPASS, GRUNDBUCH AUSZUG, FLURKARTE,
ENERGIE AUSWEIS, KAUFVERTRAG, FINANZIERUNGSBESTÄTIGUNG,
TESTAMENT, GESELLSCHAFTSVERTRAG, HANDELSREGISTER AUSZUG, ...
```

4.4 Finanzen

```
@dataclass
class FinancingOffer:
    offer_id: str
    finanzierer_id: str
    projekt_id: str

    # Konditionen
    darlehensbetrag: float
    zinssatz: float
    sollzinsbindung: int # Jahre
    tilgungssatz: float
    monatliche_rate: float
    effektivzins: float

    # Optionen
    sondertilgung_prozent: float
    bereitstellungszinsen_frei_monate: int

    # Status
    status: str # GESENDET, ANGENOMMEN, ABGELAUFEN
    gueltig_bis: Optional[datetime]
```

```

    fuer_notar_markiert: bool

    pdf_data: Optional[bytes]

@dataclass
class Finanzierungsmodell:
    modell_id: str
    name: str
    kaufpreis: float
    nebenkosten: float
    eigenkapital: float
    darlehensbetrag: float

    # Berechnung
    monatliche_rate: float
    restschuld_nach_zinsbindung: float
    gesamtlaufzeit_jahre: float
    gesamtzinsen: float
    tilgungsplan_json: str

    status: str # ENTWURF, FAVORIT, ANGENOMMEN
    quelle: str # EIGENE_BERECHNUNG, FINANZIERER_ANGEBOT

```

4.5 Notarielle Akten

```

@dataclass
class Akte:
    akte_id: str
    notar_id: str
    sachbearbeiter_id: Optional[str]

    # Aktenzeichen: z.B. "123/24-SCH-MU"
    aktennummer: int
    aktenjahr: int
    notar_kuerzel: str
    mitarbeiter_kuerzel: str
    aktenzeichen: str

    # Klassifizierung
    hauptbereich: str # AktenHauptbereich Enum
    untertyp: str # AktenTyp* Enums

    # Verknüpfungen
    projekt_id: Optional[str]
    parteien: List[Dict]
    dokument_ids: List[str]

    # Status
    status: str # AktenStatus Enum

```

```

    geschaeftswert: float
    gebuehren: float
    gebuehren_bezahlt: bool

# AktenHauptbereich Enum:
ERBRECHT, GESELLSCHAFTSRECHT, ZIVILRECHT, FAMILIENRECHT, SONSTIGE

```

4.6 Verträge & Textbausteine

```

@dataclass
class Textbaustein:
    baustein_id: str
    notar_id: str
    titel: str
    text: str
    zusammenfassung: str
    kategorie: str # TextbausteinKategorie Enum
    vertragstypen: List[str]

    # KI-Features
    ki_generiert: bool
    ki_kategorisiert: bool
    ki_update_vorschlag: str

    # Versionierung
    version: int
    vorherige_version_id: Optional[str]
    status: str # ENTWURF, FREIGEgeben, ARCHIVIERT

@dataclass
class Vertragsentwurf:
    entwurf_id: str
    projekt_id: str
    vertragstyp: str
    volltext: str
    baustein_ids: List[str]

    # Wünsche
    kaeufer_wuensche: List[str]
    verkaeufer_wuensche: List[str]

    # Status
    status: str # ENTWURF, FREIGEgeben, VERSENDET
    versendet_an: List[str]

    pdf_data: Optional[bytes]

```

4.7 DSGVO & Datenschutz

```
@dataclass
class PersonenbezogeneDaten:
    daten_id: str
    betroffener_id: str
    kategorie: str # DatenKategorie Enum
    datenfelder: List[str]
    herkunft: str # DatenHerkunft Enum
    rechtsgrundlage: str
    einwilligung_erteilt: bool
    aufbewahrungsfrist_jahre: int
    ist_geloescht: bool

@dataclass
class LoeschAnfrage:
    anfrage_id: str
    betroffener_id: str
    loeschgrund: str
    status: str # LoeschStatus Enum
    frist_bis: datetime
    protokoll_id: str

# DatenKategorie Enum:
STAMMDATEN, KONTAKTDATEN, FINANZDATEN, AUSWEISDATEN,
GESUNDHEITSDATEN, KOMMUNIKATION, DOKUMENTE, VERTRAGSDATEN
```

5. Hilfsfunktionen

5.1 GNotKG-Berechnungen (Notargebühren)

```
# Vollgebühr nach GNotKG ermitteln
get_gnotkg_vollgebuehr(geschaeftswert: float) → float

# Notarkosten für Kaufvertrag
berechne_notarkosten_kaufvertrag(kaufpreis: float) → {
    "beurkundung": float,      # 2,0-fache Gebühr
    "vollzug": float,          # 0,5-fache Gebühr
    "betreuung": float,        # 0,5-fache Gebühr
    "netto": float,
    "mwst": float,
    "brutto": float
}
```

```

# Grundbuchkosten
berechne_grundbuchkosten_kaufvertrag(kaufpreis: float) → {
    "umschreibung": float,      # 1,0-fache Gebühr
    "vormerkung": float,        # 0,5-fache Gebühr
    "gesamt": float
}

# Grundschkuldkosten
berechne_grundschuldkosten(grundschuldbetrag: float, anzahl: int) → Dict

# Löschungskosten
berechne_loeschungskosten(betrag: float, anzahl: int) → Dict

# Gesamtkosten für Käufer
berechne_gesamtkosten_kaeufer(
    kaufpreis: float,
    makler_provision_prozent: float,
    grundschulden: List,
    grunderwerbsteuer_prozent: float
) → {
    "kaufpreis": float,
    "notarkosten": float,
    "grundbuchkosten": float,
    "grundschuldkosten": float,
    "maklerkosten": float,
    "grunderwerbsteuer": float,
    "nebenkosten_gesamt": float,
    "gesamtkosten": float
}

```

5.2 Session-State-Operationen

```

# Session initialisieren (50+ Datenstrukturen)
init_session_state()

# Session-Token generieren
get_session_token(email: str) → str # SHA256 Hash

# Session im Browser speichern/laden
save_session_to_browser(email: str, token: str)
restore_session_from_storage() → Optional[User]

```

5.3 Benachrichtigungen

```

# Benachrichtigung erstellen
create_notification(
    user_id: str,

```

```

    titel: str,
    nachricht: str,
    typ: str = "INFO", # INFO, WARNING, ERROR, SUCCESS
    link: str = None
) → str # notification_id

# Ungelesene abrufen
get_unread_notifications(user_id: str) → List[Notification]

```

5.4 Suche & Filterung

```

# Generische Suche
search_matches(search_term: str, *fields) → bool

# Spezifische Filter
filter_projekte_by_search(projekte: list, search_term: str) → list
filter_dokumente_by_search(dokumente: list, search_term: str) → list
filter_angebote_by_search(angebote: list, search_term: str) → list

```

5.5 Akten-Management

```

# Nächste Aktennummer
get_naechste_aktensnummer(notar_id: str) → Tuple[int, int]

# Akte erstellen mit automatischem Aktenzeichen
create_akte(
    notar_id: str,
    hauptbereich: str,
    untermyp: str,
    verkaeufer_nachname: str = "",
    kaeufer_nachname: str = "",
    ...
) → Akte # Aktenzeichen: "123/24-XX-YY"

```

6. Kommunikationsschnittstellen

6.1 Matrix der Kommunikationswege

Von → Nach	Mechanismus	Status
Käufer ↔ Verkäufer	Preisangebote	⚠ Nur indirekt

Von → Nach	Mechanismus	Status
Käufer → Makler	Einladungen, Exposé	✓ OK
Käufer → Notar	Dokumente, Termine	✓ OK
Käufer → Finanzierer	Anfragen, Angebote	✓ OK
Verkäufer → Makler	Dokumente, Marktanalyse	✓ OK
Verkäufer → Notar	Dokumente	⚠ Einseitig
Makler → Notar	Termine, Parteien	✓ OK
Notar → Alle	Benachrichtigungen	✓ OK

6.2 Benachrichtigungs-Trigger

```
# Bei Preisangebot (Z. 7582-7608)
create_notification(verkaeufer_id, "Neues Preisangebot", ...)
create_notification(kaeufer_id, "Angebot gesendet", ...)
create_notification(makler_id, "Preisangebot eingegangen", ...)

# Bei Preiseinigung (Z. 7648-7686)
create_notification(notar_id, "Preiseinigung erzielt", ...)
# Projekt-Kaufpreis automatisch aktualisiert

# Bei Dokument-Upload (Z. 19516-19522)
create_notification(makler_id, "Neues Dokument verfügbar", ...)

# Bei Finanzierungsanfrage (Z. 16005-16010)
create_notification(finanzierer_id, "Neue Finanzierungsanfrage", ...)
```

6.3 Dokumenten-Freigaben

```
# Verkäufer-Dokument Freigaben
VerkäuferDokument:
    freigegeben_fuer_makler: bool
    freigegeben_fuer_notar: bool
    freigegeben_fuer_finanzierer: bool
    freigegeben_fuer_kaeufer: bool

# Wirtschaftsdaten Freigaben
WirtschaftsdatenDokument:
    sichtbar_fuer_makler: bool
```

```
sichtbar_fuer_notar: bool
freigegeben_fuer_notar: bool
```

7. Bekannte Einschränkungen

7.1 Kritische Einschränkungen

Problem	Beschreibung	Auswirkung
Session-State	Daten nur im RAM	Datenverlust bei Reload
ID-Generierung	<code>len()</code> statt UUID	Duplikate möglich
Keine DB	Kein persistenter Speicher	Multi-User nicht möglich
Keine Echtzeit	Kein WebSocket/Polling	Manuelle Aktualisierung nötig

7.2 Fehlende Validierungen

```
# Preisangebote: 0€ erlaubt
angebot_betrag = st.number_input("Angebot", min_value=0.0) # FEHLER

# Eigenkapital: Kann größer als Kaufpreis sein
eigenkapital = st.number_input("Eigenkapital", min_value=0.0) # FEHLER

# Dokumenten-Freigaben: Default auf True
freigabe_makler = st.checkbox("Für Makler", value=True) # RISIKO
```

7.3 Fehlende Features

Feature	Priorität	Beschreibung
Direkte Nachrichten	HOCH	Käufer ↔ Verkäufer
Aufgaben für Verkäufer	MITTEL	Nur bei Käufer vorhanden
Audit-Logging	HOCH	Für Compliance
E-Mail-Versand	MITTEL	Nur simuliert
Persistenz	KRITISCH	PostgreSQL/MongoDB

7.4 Empfohlene Verbesserungen

1. **Datenbank-Migration:** Session-State → PostgreSQL
 2. **UUID für IDs:** `uuid.uuid4()` statt `len()`
 3. **Validierung:** Pydantic für Input-Validierung
 4. **RBAC:** Konsistente Berechtigungsprüfung
 5. **Audit-Log:** Alle kritischen Operationen loggen
-

Anhang: Enum-Referenz

Benutzer

- `UserRole` : KAEUFER, VERKAEUFER, MAKLER, NOTAR, FINANZIERER, ADMIN

Projekte

- `ProjektStatus` : VORBEREITUNG → ABGESCHLOSSEN (10 Status)
- `PropertyType` : WOHNUNG, HAUS, GRUNDSTUECK, GEWERBE, ...
- `GrundschuldStatus` : NICHT_BEGONNEN → GELOESCHT

Termine

- `TerminTyp` : BESICHTIGUNG, BEURKUNDUNG, UEBERGABE, ...
- `TerminStatus` : VORGESCHLAGEN → ABGESCHLOSSEN

Dokumente

- `DokumentTyp` : 40+ Typen (PERSONALAUSWEIS bis GESELLSCHAFTSVERTRAG)
- `DocumentRequestStatus` : ANGEFORDERT, BEREITGESTELLT, ABGELEHNT

Workflow

- `WorkflowStepStatus` : OPEN, STARTED, COMPLETED, BLOCKED
- `AktenStatus` : NEU → ARCHIVIERT (11 Status)

DSGVO

- `DatenKategorie` : STAMMDATEN, FINANZDATEN, AUSWEISDATEN, ...

- **LoeschStatus** : ANGEFRAGT → ABGELEHNT

Dokumentation erstellt am 22.12.2025