

System Specification: Universal Vehicle Identification & Origin Decoding Engine (UVI-ODE)

1. System Scope & Regulatory Standards Matrix

The Universal Vehicle Identification & Origin Decoding Engine (UVI-ODE) is an engine designed to deterministically resolve the manufacturer, geographic origin, and physical assembly plant of motor vehicles across all international jurisdictions.

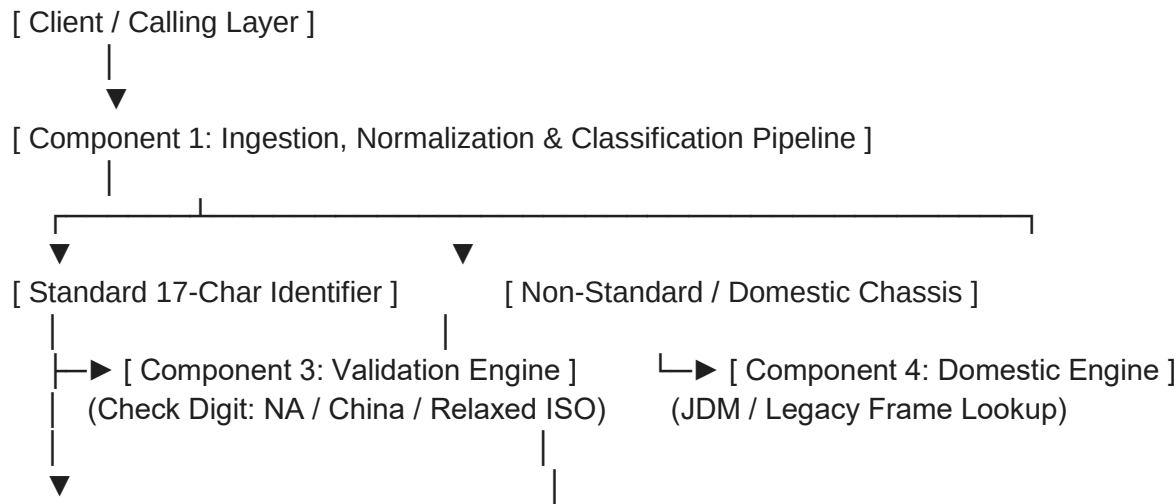
The engine evaluates input strings against the regulatory frameworks outlined in the matrix below:

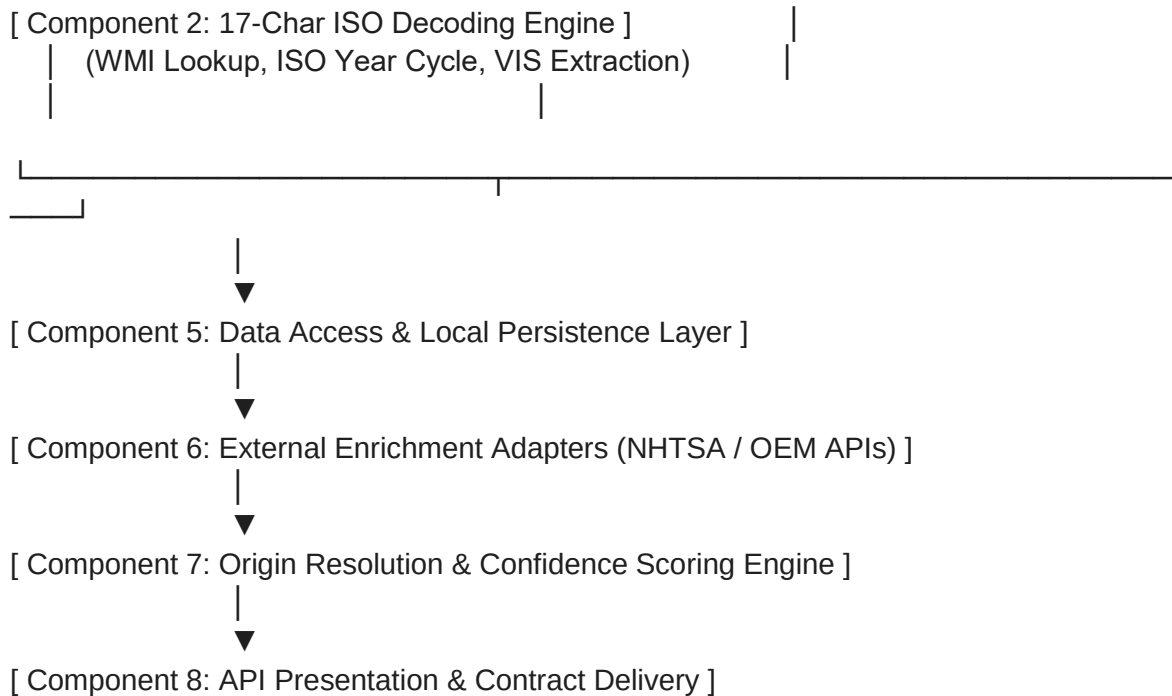
Jurisdiction / Market	Governing Regulatory Standard	String Length & Type	Mandatory Check Digit	Plant Identification Mechanism
Global / International	ISO 3779:2009, ISO 3780:2009	17 Characters (Alphanumeric, excl. I, O, Q)	No (Position 9 is manufacturer-discretionary or VDS)	Position 11 (VIS Position 2) — Manufacturer assigned
North America (USA/Canada)	49 CFR Part 565 (NHTSA), CMVSS 115 (Transport Canada)	17 Characters (Alphanumeric, excl. I, O, Q)	Yes (Position 9, Modulo 11 with weights [8\dots2]\$)	Position 11 (VIS Position 2) — Registered plant code filed with agency
China	GB 16735-2004, GB 7258-2017	17 Characters (Alphanumeric, excl. I, O, Q)	Yes (Position 9, identical algorithm to North America)	Position 11 (VIS Position 2) — MIIT filed plant code
European Union	Regulation (EU) 2019/2144, ISO 3779	17 Characters (Alphanumeric, excl. I, O, Q)	No (Pos 9 frequently represents internal	Position 11 (VIS Position 2) — Plant-specific

			descriptor)	
Australia / New Zealand	ADR 61/02, ISO 3779	17 Characters (Alphanumeric, excl. I, O, Q)	No (Permissive ISO rules)	Position 11 — Manufacturer defined
Japan Domestic Market (JDM)	MLIT Road Transport Vehicle Act (Chassis/Frame No.)	Variable (9–14 Characters, e.g., FD3S-102345, CBA-R35-012345)	No (Chassis format: Model Code + - + Serial)	Model prefix determines assembly series; Country = JPN (100% domestic assembly)
Pre-1981 Legacy (Global)	Non-standardized manufacturer serials	Variable (5–14 Characters, e.g., 136379K123456)	No	Non-standard; requires manufacturer build-sheet lookup

2. Architecture & Component Decomposition

The system is organized into decoupled, isolated components with strict interface boundaries:





3. Component 1: Ingestion, Normalization & Classification Pipeline

3.1 Input Specifications & Preprocessing

Every input string must pass through the following deterministic transformation pipeline before reaching any decoding engine:

1. **Encoding Sanitization:** Convert string to UTF-8. Strip zero-width spaces (`\u200B`), non-breaking spaces (`\u00A0`), control characters, and leading/trailing whitespace.
2. **Casing Normalization:** Transform all characters to ASCII uppercase (`a-z` `$\to$` `A-Z`).
3. **Delimiter Handling:**
 - If length `$> 17$` and contains hyphens or spaces, strip hyphens (`-`) and spaces ().
 - If length `$\le 14$` and contains exactly one hyphen, retain the hyphen for JDM Chassis processing.

3.2 Classification Rules (Regex Routing)

Pipeline Classifier:

IF Input MATCHES `Regex_ISO_Standard`:

Route to Component 3 (Validation) -> Component 2 (ISO Engine)

ELSE IF Input MATCHES `Regex_JDM_Chassis`:

Route to Component 4 (Domestic Engine)

ELSE IF Input MATCHES `Regex_Pre1981_Legacy`:

Route to Component 4 (Domestic Engine: Legacy Mode)
 ELSE:
 Throw MALFORMED_IDENTIFIER_ERROR (Code: 4001)

- **Regex_ISO_Standard:**
 $\wedge[A-HJ-NPR-Z0-9]\{17\}\$$
(Strict exclusion of I, O, Q to prevent optical confusion with 1 and 0)
- **Regex_JDM_Chassis:**
 $\wedge[A-Z0-9]\{2,7\}-[0-9]\{5,8\}\$$
(Captures all certified Japanese Domestic chassis marks: e.g., E-BCNR33-001234, ZN6-012345)
- **Regex_Pre1981_Legacy:**
 $\wedge[A-Z0-9]\{5,15\}\$$
(Alphanumeric strings that fail ISO length and lack the JDM hyphen)

4. Component 2 & 3: Standard 17-Character ISO Engine & Validation

4.1 Check Digit Mathematical Verification (Position 9)

Mandatory for North American (WMI starting with 1, 2, 3, 4, 5) and Chinese (WMI starting with L) vehicles. Permissive for ISO/EU vehicles.

Transliteration Value Map (\$V\$)

Letters are mapped to numerical values:

$$\begin{array}{l} A=1 \ \& \ B=2 \ \& \ C=3 \ \& \ D=4 \ \& \ E=5 \ \& \ F=6 \ \& \ G=7 \ \& \ H=8 \ \backslash \ J=1 \ \& \ K=2 \ \& \ L=3 \ \& \\ M=4 \ \& \ N=5 \ \& \ P=7 \ \& \ R=9 \ \& \ S=2 \ \& \ T=3 \ \& \ U=4 \ \& \ V=5 \ \& \ W=6 \ \& \ X=7 \ \& \ Y=8 \ \& \ Z=9 \end{array}$$

Digits \$0\$ through \$9\$ retain their literal values (\$0 \dots 9\$).

Weight Array (\$W\$)

Positional weights for indices \$i = 1\$ to \$17\$:

$$W = [8, 7, 6, 5, 4, 3, 2, 10, 0, 9, 8, 7, 6, 5, 4, 3, 2]$$

Algorithm

1. Compute the weighted sum:

$$S = \sum_{i=1}^{17} (V(\text{VIN}[i]) \times W[i])$$
2. Compute the remainder:

$$R = S \bmod{11}$$
3. Resolve the expected character:

$$\text{Expected} = \begin{cases} \text{"X"}, & \text{if } R = 10 \\ \text{str}(R), & \text{if } R < 10 \end{cases}$$
4. Compare \$\text{Expected}\$ with \$\text{VIN}[9]\$:
 - **If match:** Flag check_digit_valid = TRUE.
 - **If mismatch AND WMI indicates NA/China:** Flag check_digit_valid = FALSE, error status INVALID_CHECKSUM.
 - **If mismatch AND WMI indicates Europe/Rest-of-World:** Flag check_digit_valid =

NULL, classification ISO_NON_CHECK_DIGIT_COMPLIANT. Do not abort execution.

4.2 ISO 3780 Geographic Character Table (Positions 1 and 2)

The geographic region and country are resolved by comparing positions 1 and 2 against the following international allocation:

[A-H] AFRICA

AA-AH: South Africa	CA-CE: Benin	EA-EE: Ethiopia
AJ-AN: Ivory Coast	CF-CK: Madagascar	EF-EK: Mozambique
BA-BE: Angola	CL-CR: Tunisia	FA-FE: Ghana
BF-BK: Kenya	DA-DE: Egypt	FF-FK: Nigeria
BL-BR: Tanzania	DF-DK: Morocco	FL-FR: Madagascar
DL-DR: Zambia		

[J-R] ASIA

JA-JT: Japan	MA-ME: India	PA-PE: Philippines
KL-KR: South Korea	MF-MK: Indonesia	PF-PK: Singapore
L0-L9, LA-LZ: China	ML-MR: Thailand	PL-PR: Malaysia
MS-M0: Myanmar		RA-RE: UAE
NA-NE: Iran		RF-RK: Taiwan
NF-NK: Pakistan		RL-RR: Vietnam
NL-NR: Turkey		RS-R0: Saudi Arabia

[S-Z] EUROPE

SA-SM: United Kingdom	TJ-TP: Czech Republic	VA-VE: Austria
SN-ST: Germany (East)	TR-TV: Hungary	VF-VR: France
SU-SZ: Poland	TW-T1: Portugal	VS-VW: Spain
S1-S4: Latvia	UH-UM: Denmark	VX-V2: Serbia / Croatia
TA-TH: Switzerland	UN-UT: Ireland	WA-W0: Germany
UU-UZ: Romania		XA-XE: Bulgaria
U5-U7: Slovakia		XF-XK: Greece
XL-XR: Netherlands		YS-YW: Sweden
XS-XW: Russia / USSR		ZA-ZR: Italy
YA-YE: Belgium		ZX-Z2: Slovenia
YF-YK: Finland		Z3-Z5: Lithuania

[1-5] NORTH AMERICA

1A-10: United States	3A-3W: Mexico
2A-20: Canada	3X-37: Costa Rica
4A-40: United States	38-30: Cayman Islands
5A-50: United States	

[6-7] OCEANIA

6A-6W: Australia 7A-7E: New Zealand

[8-9] SOUTH AMERICA

8A-8E: Argentina 8X-8Z: Venezuela 9F-9K: Colombia
 8F-8K: Chile 9A-9E: Brazil 9L-9R: Paraguay
 8L-8R: Ecuador 9S-9W: Uruguay 93-99: Brazil
 8S-8W: Peru

4.3 Low-Volume Manufacturer Rule (Position 3 = 9)

- If character 3 is '9', the vehicle is produced by a low-volume manufacturer (\$<500\$ units/year).
- In this condition, the true manufacturer identifier is determined by combining positions 1–3 with positions 12, 13, and 14 (WMI_EXTENDED = VIN[1..3] + VIN[12..14]).
- The system must query the WMI database using the 6-character composite identifier to identify the manufacturer.

4.4 Model Year Cycle Resolution (Position 10)

ISO 3779 specifies a 30-year character cycle for position 10 (excluding I, O, Q, U, Z).

Co de	Cyc le 1	Cyc le 2	Co de	Cyc le 1	Cyc le 2	Co de	Cyc le 1	Cyc le 2	Co de	Cyc le 1	Cyc le 2
A	1980	2010	L	1990	2020	Y	2000	2030	7	2007	2037
B	1981	2011	M	1991	2021	1	2001	2031	8	2008	2038
C	1982	2012	N	1992	2022	2	2002	2032	9	2009	2039
D	1983	2013	P	1993	2023	3	2003	2033			
E	1984	2014	R	1994	2024	4	2004	2034			
F	1985	2015	S	1995	2025	5	2005	2035			

G	198 6	201 6	T	199 6	202 6	6	200 6	203 6			
H	198 7	201 7	V	199 7	202 7						
J	198 8	201 8	W	199 8	202 8						
K	198 9	201 9	X	199 9	202 9						

Disambiguation Rule: Position 7 distinguishes the cycle. In passenger cars complying with North American regulations, if position 7 is a letter, the year falls in Cycle 2 (2010–2039). If position 7 is numeric, the year falls in Cycle 1 (1980–2009). For non-NA vehicles, external enrichment is required to lock the epoch.

5. Component 4: Domestic & Non-Standard Chassis Engine (JDM & Legacy)

5.1 Japanese Domestic Market (JDM) Processing Engine

When input satisfies `Regex_JDM_Chassis`, standard 17-character validation is bypassed:

1. **Extraction:**
 - Split string by `-`.
 - Left token = `Model_Prefix` (e.g., E-BCNR33, FD3S, ZN6).
 - Right token = `Production_Serial` (e.g., 001234).
2. **Deterministic Geography:**
 - Country of Assembly: **Japan** (JPN).
 - Geographic Region: **Asia** (ASI).
 - Source: **MLIT Type Designation Certificate Register**.
 - Confidence Level: **High (Deterministic)**.
3. **Manufacturer Mapping Table (Model Prefix Index):**

Prefix Token Regex	Manufacturer Identity	Common Models	Domestic Plant Origin
<code>^(DBA-)?ZN6.*</code>	Toyota / Subaru JV	86 / BRZ	Gunma Main Plant, Ota, Japan

^(E-	GF-)?FD3S.*`	Mazda Motor Corporation	RX-7
^(E-	GF-)?BCNR33.*`	Nissan Motor Co., Ltd.	Skyline GT-R
^(ABA-)?AP[12].*	Honda Motor Co., Ltd.	S2000	Takanezawa / Suzuka, Japan
^(CBA-)?R35.*	Nissan Motor Co., Ltd.	GT-R	Tochigi Plant, Japan
^(GH-)?CT9A.*	Mitsubishi Motors	Lancer Evolution	Mizushima Plant, Japan

6. Component 5: Data Access & Local Persistence Layer

The database schema (PostgreSQL 14+ / SQLite 3.38+) stores the core WMI tables, plant overrides, and external cache entries:

-- Table: Geographic Regions (ISO 3166-1 alpha-3 compliant)

```
CREATE TABLE geo_countries (
  iso_alpha2 CHAR(2) PRIMARY KEY,
  iso_alpha3 CHAR(3) NOT NULL UNIQUE,
  country_name VARCHAR(100) NOT NULL,
  continent_code CHAR(2) NOT NULL
);
```

-- Table: WMI Registry

```
CREATE TABLE wmi_registry (
  wmi_code VARCHAR(3) NOT NULL,
  is_low_volume BOOLEAN DEFAULT FALSE,
  wmi_extended VARCHAR(3) NULL, -- For Pos 12-14 when is_low_volume = TRUE
  manufacturer_name VARCHAR(150) NOT NULL,
```

```

country_code CHAR(2) NOT NULL REFERENCES geo_countries(iso_alpha2),
vehicle_type VARCHAR(50) NOT NULL,
PRIMARY KEY (wmi_code, COALESCE(wmi_extended, '000'))
);
CREATE INDEX idx_wmi_code ON wmi_registry(wmi_code);

-- Table: Known Assembly Plants (Internal Dictionary)
CREATE TABLE assembly_plants (
  id SERIAL PRIMARY KEY,
  wmi_code VARCHAR(3) NOT NULL,
  plant_code CHAR(1) NOT NULL,
  plant_name VARCHAR(150) NOT NULL,
  city VARCHAR(100) NOT NULL,
  state_province VARCHAR(100) NULL,
  country_code CHAR(2) NOT NULL REFERENCES geo_countries(iso_alpha2),
  active_from_year INT NULL,
  active_to_year INT NULL,
  UNIQUE (wmi_code, plant_code)
);
CREATE INDEX idx_assembly_plant_lookup ON assembly_plants(wmi_code, plant_code);

-- Table: VIN Cache & Query Audit
CREATE TABLE vin_cache (
  vin_normalized VARCHAR(17) PRIMARY KEY,
  input_format VARCHAR(20) NOT NULL, -- 'ISO_17', 'JDM_FRAME', 'LEGACY'
  wmi_country_code CHAR(2) NOT NULL,
  actual_plant_country_code CHAR(2) NULL,
  plant_city VARCHAR(100) NULL,
  confidence_level VARCHAR(20) NOT NULL, -- 'HIGH_DETERMINISTIC', 'HIGH_ENRICHED',
'MODERATE_WMI_ONLY', 'UNKNOWN'
  raw_api_payload JSONB NULL,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
  expires_at TIMESTAMP WITH TIME ZONE NOT NULL
);
CREATE INDEX idx_vin_cache_expiry ON vin_cache(expires_at);

```

7. Component 6: External Enrichment Adapters

When an ISO 17-character VIN is decoded locally, the Assembly Plant code (Position 11) is ambiguous unless matched against known manufacturer records. The external enrichment adapter resolves Position 11 dynamically.

7.1 Primary Open Adapter: NHTSA vPIC Endpoint

- **Base URL:**
[\[https://vpic.nhtsa.dot.gov/api/vehicles/DecodeVinValues/\]](https://vpic.nhtsa.dot.gov/api/vehicles/DecodeVinValues/)(<https://vpic.nhtsa.dot.gov/api/vehicles/DecodeVinValues/>){VIN}?format=json

- **Transport:** HTTPS GET, Timeout: 2500ms, Retries: 1 with exponential backoff.
- **Authentication:** None (Public Domain, US Federal Agency).
- **Extraction Field Path:**
 - Actual Assembly Country: Results[0].PlantCountry
 - Actual Assembly City: Results[0].PlantCity
 - Manufacturer Name: Results[0].Make / Results[0].Manufacturer
 - Error Check: Results[0].ErrorCode (Must be "0" or evaluate to non-fatal)

7.2 Secondary Commercial Adapter Specification (Fallback Interface)

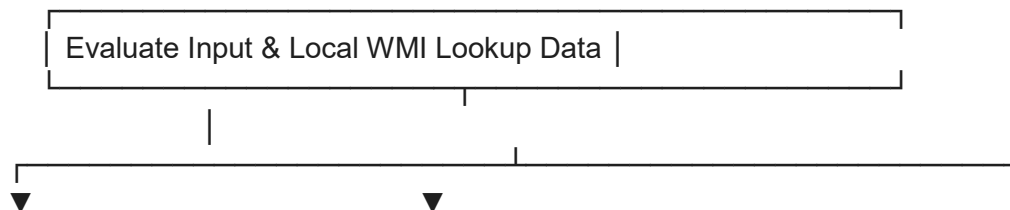
To support commercial enterprise upgrades (e.g., DataOne, Carfax, or AutoCheck) without altering domain logic, external adapters implement the following interface contract:

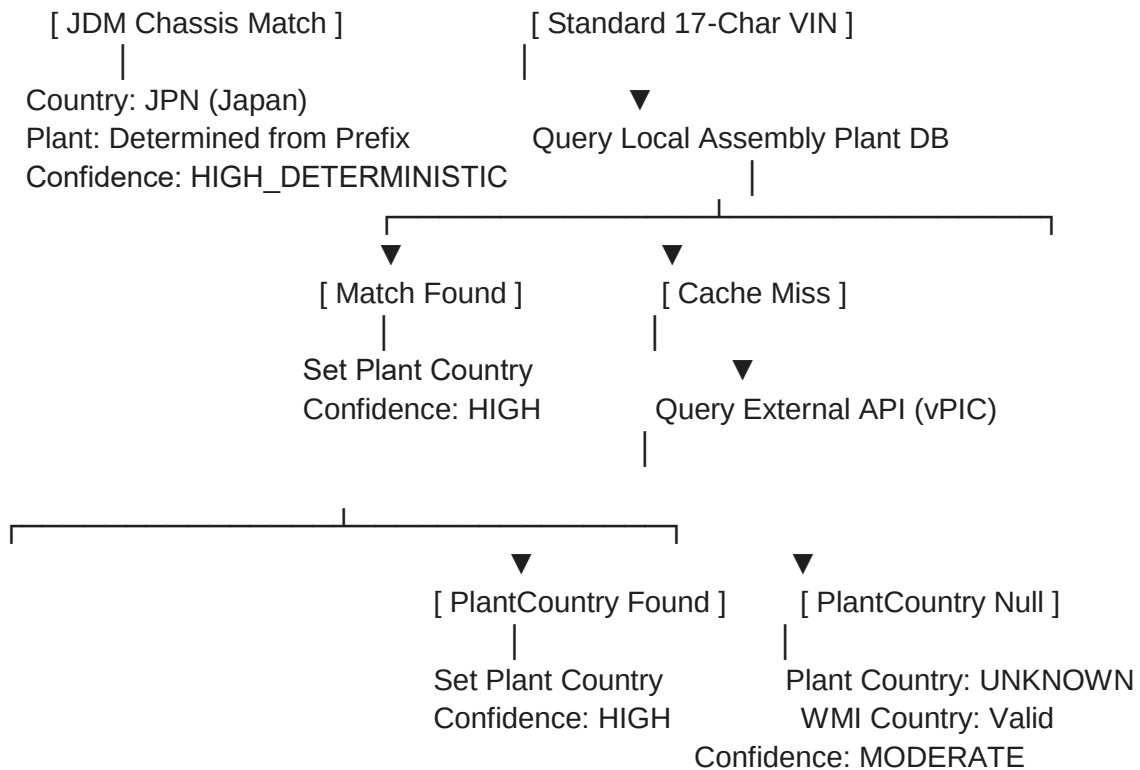
```
interface ExternalEnrichmentProvider {
  decodeAssembly(vin: string): Promise<EnrichmentResult>;
}
```

```
interface EnrichmentResult {
  isSuccessful: boolean;
  actualPlantCountryIso2: string | null;
  actualPlantCity: string | null;
  manufacturerName: string | null;
  rawPayload: Record<string, unknown>;
  errorCode: string | null;
}
```

8. Component 7: Origin Resolution & Confidence Scoring Engine

The origin resolution engine executes the following arbitration matrix to resolve the difference between the corporate WMI Country and the actual Physical Assembly Plant:





8.1 Confidence Scoring Rules

1. **HIGH_DETERMINISTIC:**
 - Evaluated for JDM frame codes successfully verified against the chassis index.
 - Or 17-character VIN where the assembly plant code at Position 11 is definitively resolved via local validated plant database or government filing.
2. **HIGH_ENRICHED:**
 - Resolved via external API adapter with matching validation status.
3. **MODERATE_WMI_ONLY:**
 - WMI is valid and maps to a recognized country, but the specific plant at Position 11 cannot be resolved from either local databases or external adapters.
 - **System Output Rule:** The system outputs the WMI manufacturer country and explicitly flags the assembly plant as "UNVERIFIED_ASSEMBLY_LOCATION".
4. **LOW_INSUFFICIENT_DATA:**
 - Pre-1981 non-standard VIN, or WMI country is resolved but check digit fails on North American/Chinese registrations.
5. **INVALID:**
 - Unparseable characters, invalid lengths, illegal characters (I, O, Q).

9. Component 8: API Layer & Error Catalog

9.1 RESTful Endpoint Specification

- **Route:** POST /api/v1/vin/decode
- **Content-Type:** application/json

Request Payload

```
{
  "identifier": "4T1B11HK5EU123456",
  "options": {
    "allow_relaxed_iso": true,
    "force_refresh_cache": false
  }
}
```

Response Payload (Assembly Mismatch Example)

```
{
  "status": "SUCCESS",
  "data": {
    "input_identifier": "4T1B11HK5EU123456",
    "format_classification": "ISO_17_STANDARD",
    "validation": {
      "is_valid": true,
      "check_digit_calculated": "5",
      "check_digit_actual": "5",
      "check_digit_status": "MATCH"
    },
    "origin": {
      "wmi_registered_country": {
        "iso_alpha2": "US",
        "iso_alpha3": "USA",
        "name": "United States",
        "region": "North America"
      },
      "actual_assembly_plant": {
        "is_verified": true,
        "country": {
          "iso_alpha2": "US",
          "iso_alpha3": "USA",
          "name": "United States"
        }
      }
    }
  }
}
```

```

    "city": "Georgetown",
    "state_province": "Kentucky",
    "plant_identifier_character": "U"
  },
  "origin_discrepancy_detected": false
},
"vehicle_attributes": {
  "wmi": "4T1",
  "manufacturer": "Toyota Motor Manufacturing, Kentucky, Inc.",
  "model_year": 2014,
  "is_low_volume_manufacturer": false
},
"meta": {
  "confidence_level": "HIGH_ENRICHED",
  "evidence_sources": [
    "LOCAL_WMI_DATABASE_V1",
    "NHTSA_VPIC_API"
  ],
  "resolved_at": "2026-10-09T14:52:00Z"
}
}
}

```

Response Payload (JDM Example)

```

{
  "status": "SUCCESS",
  "data": {
    "input_identifier": "ZN6-012345",
    "format_classification": "JDM_CHASSIS_NUMBER",
    "validation": {
      "is_valid": true,
      "check_digit_status": "NOT_APPLICABLE"
    },
    "origin": {
      "wmi_registered_country": {
        "iso_alpha2": "JP",
        "iso_alpha3": "JPN",
        "name": "Japan",
        "region": "Asia"
      }
    }
  }
}

```

```

    },
    "actual_assembly_plant": {
      "is_verified": true,
      "country": {
        "iso_alpha2": "JP",
        "iso_alpha3": "JPN",
        "name": "Japan"
      },
      "city": "Ota",
      "state_province": "Gunma",
      "plant_identifier_character": null
    },
    "origin_discrepancy_detected": false
  },
  "vehicle_attributes": {
    "manufacturer": "Subaru / Toyota JV",
    "model_designation": "Toyota 86 / Scion FR-S",
    "model_year": null
  },
  "meta": {
    "confidence_level": "HIGH_DETERMINISTIC",
    "evidence_sources": [
      "MLIT_JDM_CHASSIS_REGISTRY"
    ],
    "resolved_at": "2026-10-09T14:52:00Z"
  }
}

```

9.2 Error Code Registry

Error Code	HTTP Status	Error Type Identifier	Triggering Condition
4001	400	MALFORMED_IDENTIFIER	Input string contains invalid characters (e.g., I, O, Q in ISO mode) or invalid length (\$<5\$ or \$>17\$).
4002	422	CHECKSUM_FAIL	Position 9 check

		ED	digit recalculation failed on a mandatory market (US/Canada/China) .
4004	404	UNKNOWN_WMI	Characters 1–3 do not match any allocated ISO 3780 manufacturer country block.
5002	502	UPSTREAM_ADAPTER_ERROR	External enrichment service returned 5xx status or connection timed out after retries.

10. Core Decoding Implementation Logic (Pseudocode Reference)

This pseudocode specifies the business logic for the engine's coordinator service:

```
class UniversalVinDecoder:
    def decode(self, raw_input: str) -> bool:
        # Step 1: Normalize
        clean_str = self.normalize(raw_input)

        # Step 2: Route by Classification
        if Regex_ISO_Standard.match(clean_str):
            return self.process_iso_17(clean_str, allow_relaxed_iso)
        elif Regex_JDM_Chassis.match(clean_str):
            return self.process_jdm(clean_str)
        elif Regex_Pre1981_Legacy.match(clean_str):
            return self.process_legacy(clean_str)
        else:
            raise ValidationError(code=4001, message="Malformed identifier")
```

```

def process_iso_17          str          bool
    # Check digit verification
    is_na_or_china = vin[0] in ['1', '2', '3', '4', '5', 'L']
    check_digit_valid = self.verify_check_digit(vin)

    if is_na_or_china and not check_digit_valid:
        raise ChecksumError(code=4002, message="Invalid North American or Chinese check
digit")

    # WMI Lookup
    wmi_code = vin[0:3]
    if vin[2] == '9': # Low volume manufacturer
        wmi_extended = vin[11:14]
        wmi_record = Database.lookup_wmi(wmi_code, wmi_extended)
    else:
        wmi_record = Database.lookup_wmi(wmi_code, None)

    if not wmi_record:
        # Fallback: extract generic country from ISO 3780 position 1-2 block
        wmi_record = Iso3780Table.lookup_country(vin[0:2])

    # Position 11 Assembly Plant Extraction
    plant_char = vin[10]
    plant_record = Database.lookup_plant(wmi_code, plant_char)

    if plant_record:
        return BuildResult(wmi=wmi_record, plant=plant_record,
confidence="HIGH_DETERMINISTIC")

    # External Enrichment Fallback
    enrichment = ExternalEnricher.fetch_nhtsa(vin)
    if enrichment.has_assembly_plant:
        return BuildResult(wmi=wmi_record, plant=enrichment.plant,
confidence="HIGH_ENRICHED")

    # Moderate fallback: WMI known, assembly plant unverified
    return BuildResult(
        wmi=wmi_record,
        plant=None,
        plant_status="UNVERIFIED_ASSEMBLY_LOCATION",
        confidence="MODERATE_WMI_ONLY"
    )

```

11. Verification & Testing Matrix

The implementation will be verified against the following test suite:

[TEST-01: US Assembly, Matching WMI]

Input: 1FA6P8CF5H5XXXXXX (Ford Mustang)

Expected: WMI Country = USA, Assembly Plant = Flat Rock, USA. Check Digit = Valid.

[TEST-02: Transnational Assembly Discrepancy]

Input: 3VWDP7AJ9HMXXXXXX (Volkswagen Jetta)

Expected: WMI Country = Mexico (3VW), Assembly Plant = Puebla, Mexico. Check Digit = Valid.

Discrepancy: False (VW German brand, but WMI correctly shows Mexican subsidiary).

[TEST-03: Domestic Brand Assembled Abroad]

Input: 2T3C1RFV9MCXXXXXX (Toyota RAV4)

Expected: WMI Country = Canada (2T3), Assembly Plant = Cambridge, Ontario, Canada.

Verification: Confirms Japanese brand assembled in Canada.

[TEST-04: European Relaxed Check Digit Mode]

Input: WBA3A5C50DFXXXXXX (BMW 3 Series - German Domestic Market)

Expected: Check digit validation skipped/relaxed. WMI Country = Germany.

[TEST-05: JDM Chassis Input]

Input: FD3S-102345 (Mazda RX-7)

Expected: Classification = JDM_CHASSIS_NUMBER, Assembly Country = Japan, City = Hiroshima.

[TEST-06: Low-Volume Manufacturer Identification]

Input: WP0ZZZ99ZTSXXXXXX (Pos 3 = '9')

Expected: Triggers 6-character composite lookup (WP0 + digits 12-14).