

Excelize 开源基础库助力企业信息系统协同创新

Excelize Open-source Library Empowered of Enterprise IT Systems Collaborative Innovation

阿里巴巴 技术专家, Excelize 作者 | 续日
Alibaba Group Technical Specialist, author of Excelize | XU RI



续日

GitHub: @xuri

<https://xuri.me>

软件工程师, Excelize 开源基础库作者

阿里巴巴技术专家

曾就职于百度、360 等公司

前百度 Go 语言编程委员会成员,

从事百度 Go 语言研发体系的建设工作

内容提要

业务背景：企业级应用中，办公文档数据协同的难点

应用案例：Excelize 如何助力企业信息系统的协同创新

架构设计：办公文档格式国际标准实现过程中面临的挑战的解法

总结展望：Excelize 开源数据洞察、WebAssembly 支持和未来规划

业务背景: 电子表格办公文档的应用领域广泛

日历 & 计划

日程管理
计划任务
轮班时间表
名册模版
个人计划
每日代办

预算 & 财务

收入与支出工作表
现金流管理
预算管理
支票账单
账户系统
广告预算

销售

商机管理 CRM
内部销售
销售渠道管里
客户评价追踪
会员名录
销售业绩管理

内容生产

视频创作
播客编辑日历
媒体编辑日历
选角表和试镜表
剧组人员管理
内容营销管理

项目管理

甘特图
项目规划
敏捷看板
关键路径分析
代办列表
任务列表
时间线模版
工作项拆解

人力资源 & 工资表

员工工资
考勤记录
开支报告
报销表格
假期追踪
面试问题和反馈
人力资源知识库
员工满意度调研

研发管理

缺陷追踪
用户调研
价值风险
客户反馈
技术路线图
项目进展追踪
用户增长实验记录
产品规划表

更多领域 & 行业

金融
医疗
教育
能源
交通运输
制造业
服务业
...

企业办公文档数据协同场景

由于企业的业务特点，难以实现一套系统满足企业全场景应用

大量场景需要企业应用集成 (EAI: Enterprise Application Integration)

办公文档如同企业的“数字化细胞”，成为系统间数据协同的一种常用媒介



Excelize: 开源电子表格办公文档基础库

开源代码：完全开源的电子表格办公文档基础库 <https://github.com/xuri/excelize>

开源合规：采用 BSD 3-Clause "New" or "Revised" License 开源许可协议

开放标准：基于 ECMA-376, ISO/IEC 29500 国际标准实现

互操作性：支持读写 Excel、WPS、OpenOffice 等办公应用创建的电子表格文档

文档格式：XLAM / XLSM / XLSX / XLTM / XLTX 多种办公文档格式支持

主要特点：兼容性好、高性能、安全、功能丰富、支持跨平台

案例1: Excelize 在飞书企业协作与管理平台中的应用

借助 Excelize 基础库解析和生成电子表格中的数据，实现批量对员工信息的导入和导出，帮助组织人员信息在线化

飞书人事

人员

组织

报表

假勤

更多

花名册

入职管理

转正管理

离职管理

异动记录

花名册

在职
342

正式
205

实习
26

外包
93

其他
18

试用中
8

已转正
197

待离职
3

已离职
56

通过姓名、工作邮箱、手机号码搜索...

花名册字段

导入

导出

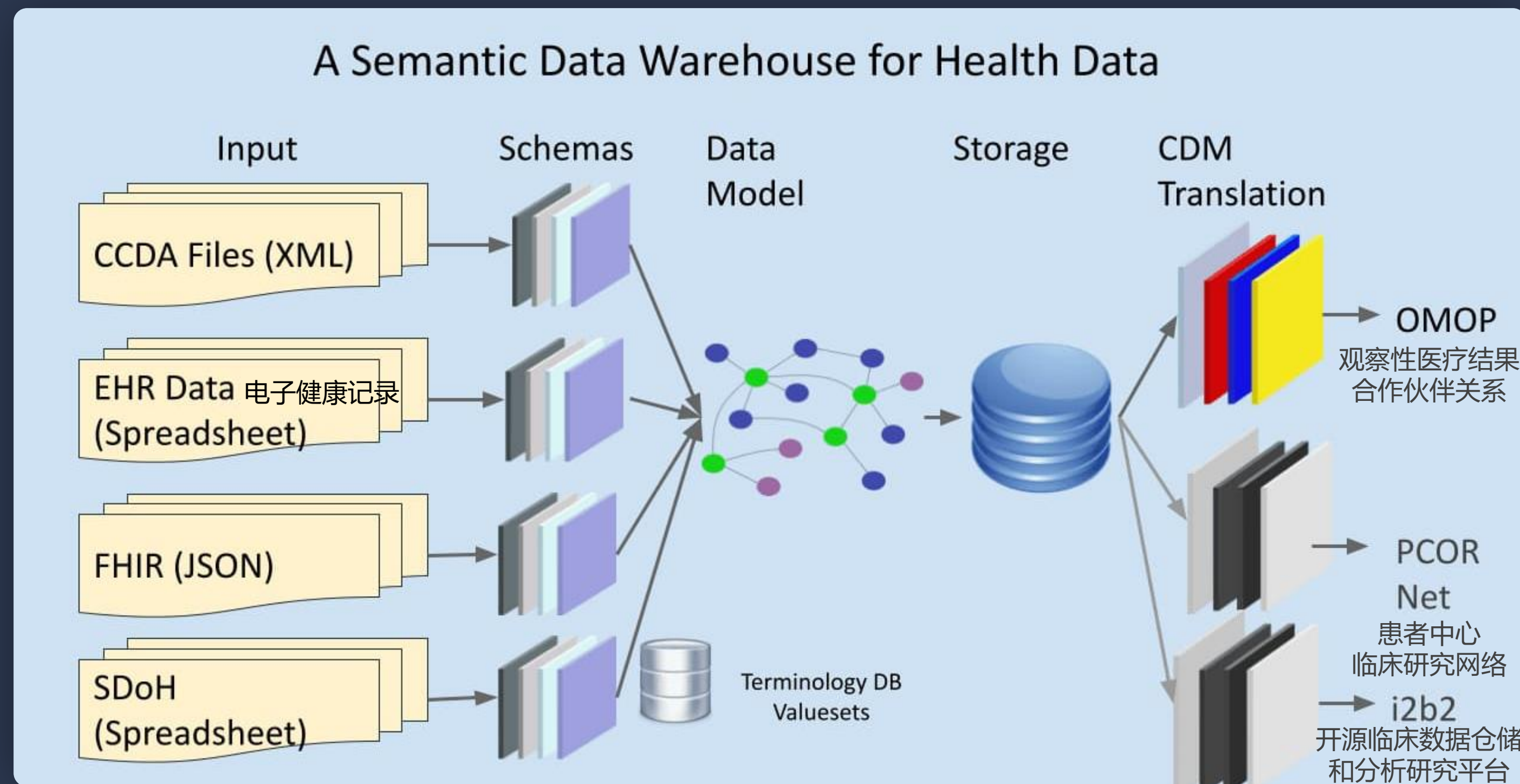
+ 添加人员

姓名	手机号码	工号	人员类型	部门	序列	操作
批量添加人员						
批量更新信息						

<https://juejin.cn/post/7117578224854368263>

案例2: Excelize 在医疗信息系统中的应用

ONC 在可分层医疗信息数据处理框架中，通过集成 Excelize 基础库对 EHR 和 SDoH 电子表格数据格式进行解析，实现异构数据源转换协调，并与 OMOP、i2b2 等系统连接协同



<https://github.com/onc-healthit/2021-LEAP-Semantic-Interoperability>

应用1: 读写电子表格文档

创建工作簿、设置单元格的值

```
f := excelize.NewFile()
// 创建一个工作表
index, _ := f.NewSheet("Sheet2")
// 设置单元格的值
f.SetCellValue("Sheet2", "A2", "Hello, world!")
f.SetCellValue("Sheet1", "B2", 100)
// 获取工作表中指定单元格的值
cell, err := f.GetCellValue("Sheet1", "B2")
if err != nil {
    fmt.Println(err)
    return
}
fmt.Println(cell)
// 设置工作簿的默认工作表
f.SetActiveSheet(index)
// 根据指定路径保存文件
if err := f.SaveAs("Book1.xlsx"); err != nil {
    fmt.Println(err)
}
```

打开工作簿、读取工作表中的单元格

```
f, err := excelize.OpenFile("Book1.xlsx")
if err != nil {
    fmt.Println(err)
    return
}
defer func() { // 关闭工作簿
    if err := f.Close(); err != nil {
        fmt.Println(err)
    }
}()
rows, err := f.GetRows("Sheet1")
if err != nil {
    fmt.Println(err)
}
for _, row := range rows {
    for _, colCell := range row {
        fmt.Print(colCell, "\t")
    }
}
```

应用2: 数据透视分析

借助 Excelize 创建数据透视表按月对各区域在
售的商品销售额进行分类汇总，并支持按销售区
域、时间和分类进行筛选

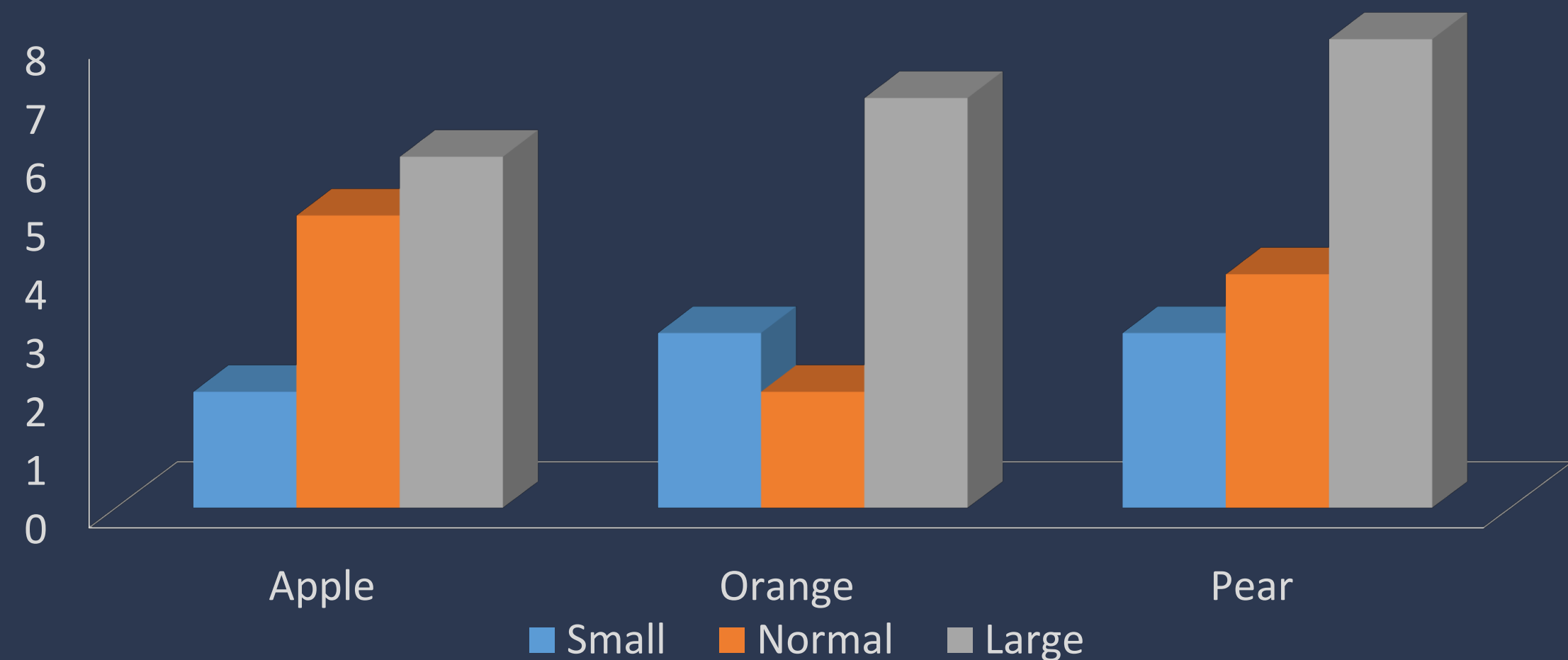
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	月	年	类型	销售额	区域		区域	(全部)					
2	6月	2017	农产品	4059	西部		销售额汇总		类型				
3	7月	2018	肉类	456	东部		月	年	农产品	肉类	饮料	乳类	总计
4	3月	2018	饮料	89	东部		6月	2017	4059				4059
5	3月	2017	乳类	3237	北部		6月汇总		4059				4059
6	12月	2017	肉类	1258	南部		7月	2018		456			456
7	8月	2018	肉类	2790	南部		7月汇总			456			456
8	2月	2017	农产品	1831	西部		3月	2017				3237	3237
9	9月	2017	农产品	1485	北部		3月汇总	2018			89		89
10	2月	2018	饮料	563	西部		12月	2017		1258			1258
11	4月	2018	肉类	1159	西部		12月汇总			1258			1258
12							8月	2018		2790			2790
13							8月汇总			2790			2790
14							2月	2017	1831				1831
15							2月	2018			563		563
16							2月汇总		1831		563		2394
17							9月	2017	1485				1485
18							9月汇总		1485				1485
19							4月	2018		1159			1159
20							4月汇总			1159			1159
21							总计		7375	5663	652	3237	16927
22													
23													
24													
25													
26													

```
f.AddPivotTable(&excelize.PivotTableOption{
    DataRange:      "Sheet1!$A$1:$E$31",
    PivotTableRange: "Sheet1!$G$2:$M$24",
    Rows: []excelize.PivotTableField{
        {Data: "Month",
         DefaultSubtotal: true},
        {Data: "Year"}},
    Filter: []excelize.PivotTableField{
        {Data: "Region"}},
    Columns: []excelize.PivotTableField{
        {Data: "Type", DefaultSubtotal: true}},
    Data: []excelize.PivotTableField{
        {Data: "Sales", Name: "销售额汇总",
         Subtotal: "Sum"}},
    RowGrandTotals: true,
    ColGrandTotals: true,
    ShowDrill:      true,
    ShowRowHeaders: true,
    ShowColHeaders: true,
    ShowLastColumn: true,
})
```

应用3: 数据可视化分析

Excelize 支持二维、三维、面积图、条形、圆锥、棱锥、柱形、堆积柱形、圆环图、折线图、饼图、雷达图、散点图、曲面图、气泡图等累计 55 种原生图表与组合图表的创建

三维簇状柱形图



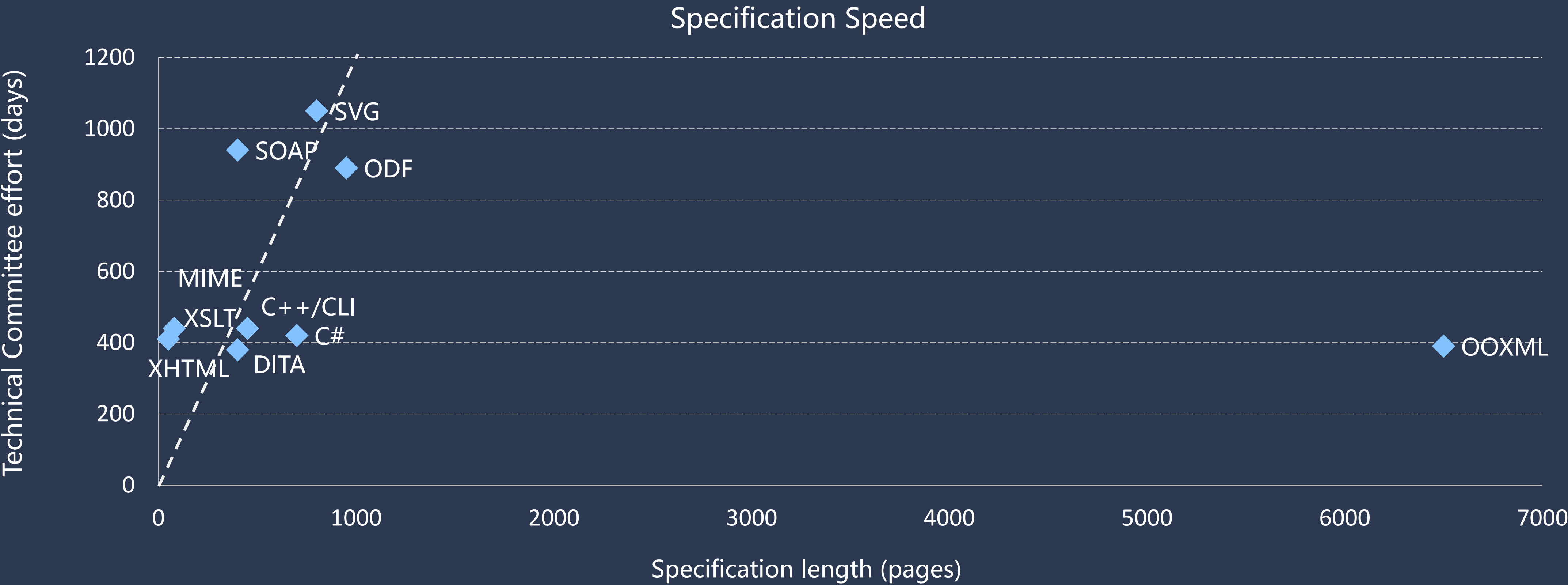
```
f.AddChart("Sheet1", "E1", &excelize.Chart{
    Type: excelize.Col3DClustered,
    Series: []excelize.ChartSeries{
        {
            Name: "Sheet1!$A$2",
            Categories: "Sheet1!$B$1:$D$1",
            Values: "Sheet1!$B$2:$D$2",
        },
        {
            Name: "Sheet1!$A$3",
            Categories: "Sheet1!$B$1:$D$1",
            Values: "Sheet1!$B$3:$D$3",
        },
        {
            Name: "Sheet1!$A$4",
            Categories: "Sheet1!$B$1:$D$1",
            Values: "Sheet1!$B$4:$D$4",
        },
    },
    Title: excelize.ChartTitle{Name: "三维簇状柱形图"},
})
```


架构设计: 相关国际标准 (部分)

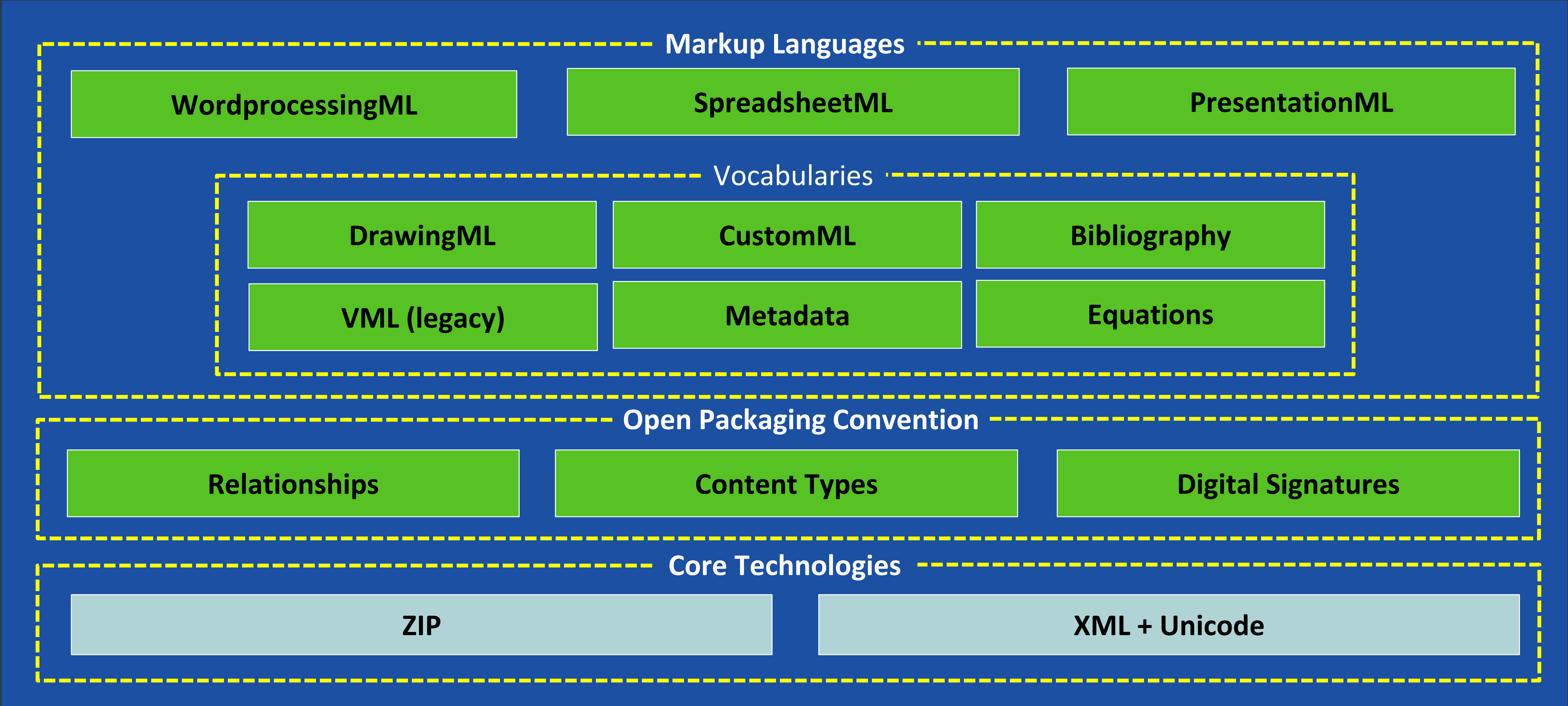
- ISO/IEC 29500 (Office Open XML File Formats)
- ISO/IEC 21320 (Document Container File)
- ISO/IEC 10646-2 (Universal Coded Character Set)
- XML (Extensible Markup Language)
- XSD (W3C XML Schema Definition Language)
- [MS-CFB]: Compound File Binary File Format
- [MS-OFFCRYPTO]: Office Document Cryptography Structure
- [MS-LCID]: Windows Language Code Identifier Reference
- DCMI (Dublin Core Metadata Initiative) Metadata Terms

标准规模对比: 庞大复杂的办公文档格式标准体系

ISO - ISO/IEC 29500-1:2016 ECMA 376 Open Office XML



标准架构

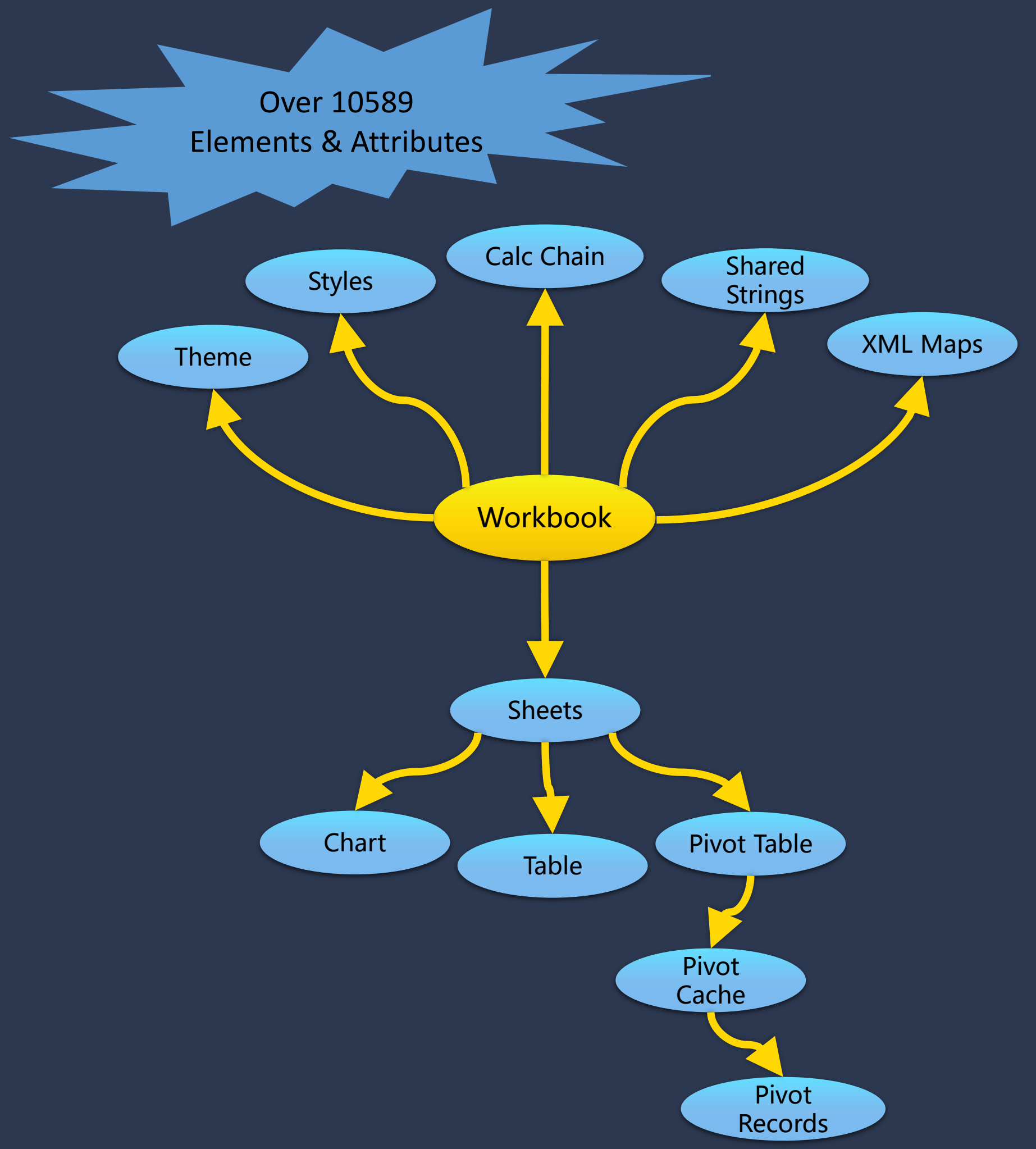


格式解析原理

B	C	D	E
		Q1	Q2
Revenue		412.52	

(build-in)

```
<c r="D3" s="7" t="s">
  <v>0</v>
</c>
<xf numFmtId="0" fontId="4" fillId="2" borderId="2" xfId="1" applyBorder="1" />
<font>
  <sz val="11"/>
  <color theme="0"/>
  <name val="Calibri"/>
  <scheme val="minor"/>
</font>
<fill>
  <patternFill patternType="solid">
    <fgColor theme="4"/>
    <bgColor theme="4"/>
  </patternFill>
</fill>
<border>
  <left style="thick">
    <color auto="1"/>
  </left>
  <right style="thick">
    <color auto="1"/>
  </right>
  <top style="thick">
    <color auto="1"/>
  </top>
  <bottom style="thick">
    <color auto="1"/>
  </bottom>
  <diagonal/>
</border>
<a:theme
  xmlns:a="http://schemas.openxmlformats.org"
  <a:themeElements>
    <a:clrScheme name="Office">
      <a:dk1>
        <a:sysClr val="windowText" />
      </a:dk1>
      <a:lt1>
        <a:sysClr val="window" />
      </a:lt1>
      <a:dk2>
        <a:srgbClr val="1F497D"/>
      </a:dk2>
      <a:lt2>
        <a:srgbClr val="FAF3E8"/>
      </a:lt2>
      <a:accent1>
        <a:srgbClr val="5C93B4"/>
      </a:accent1>
    </a:clrScheme>
  </a:themeElements>
</a:theme>
<cellStyleXfs count="12">
  <xf numFmtId="0" fontId="0" fillId="0" borderId="0"/>
  <xf numFmtId="0" fontId="4" fillId="2" borderId="0" applyBorder="1" applyAlignment="1" applyProtection="1"/>
</cellStyleXfs>
<cellStyles count="2">
  <cellStyle name="Accent1" xfId="1" builtinId="29"/>
</cellStyles>
```

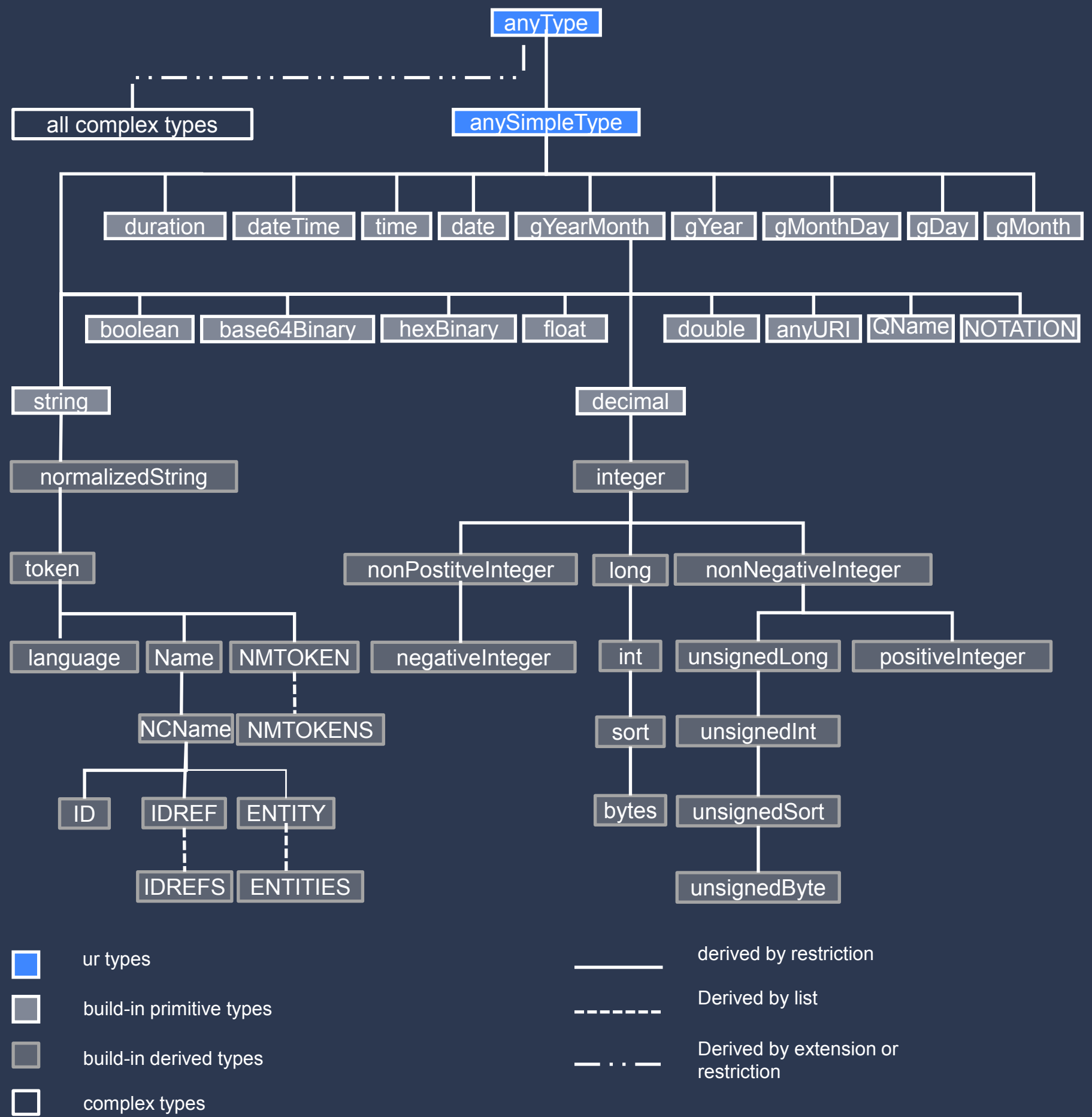
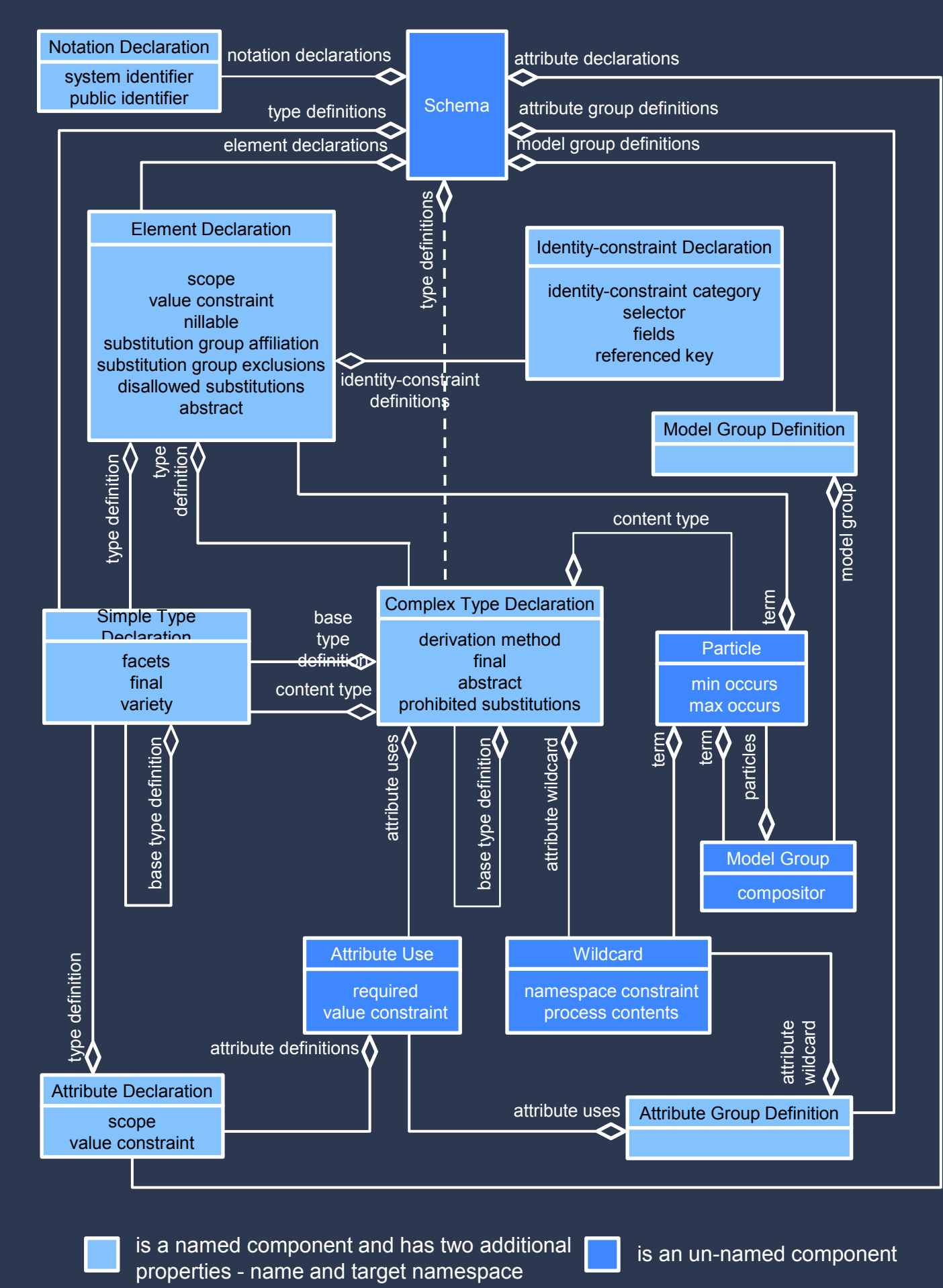


Excelize 基础库架构

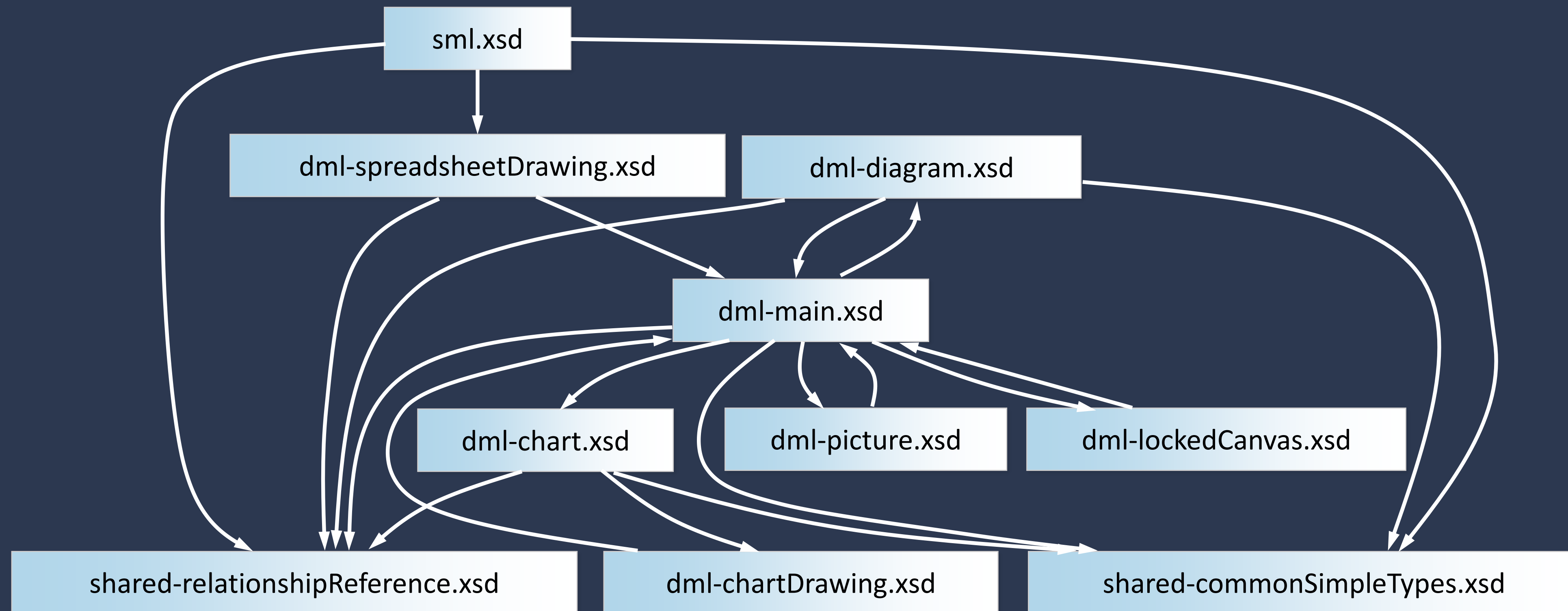


架构设计: 兼容性难题如何解决

对文档内存模型的完整度将直接影响文档的兼容性, Excelize 采用基于 XSD 定义的建模

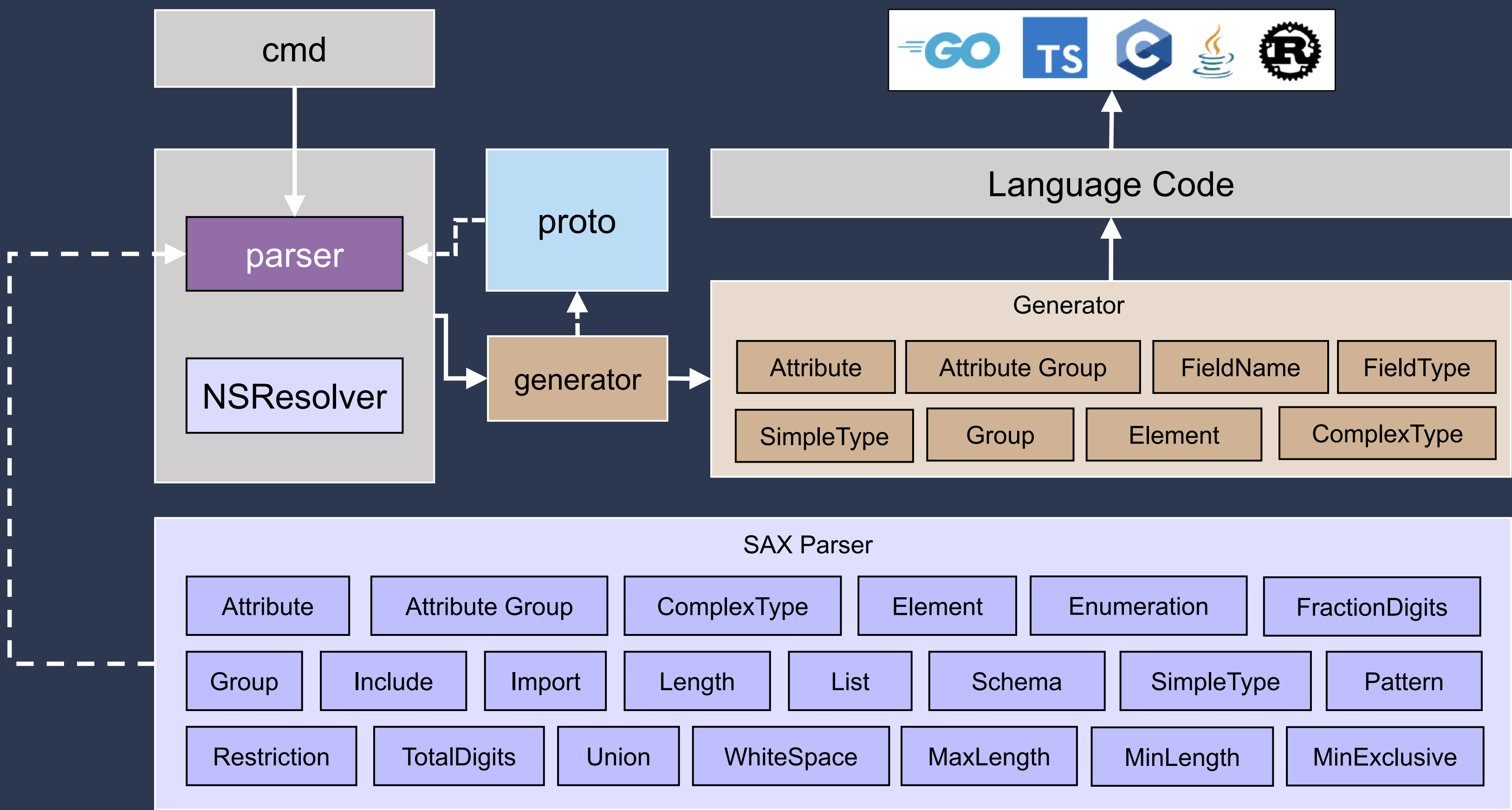


ISO/IEC 29500 中的 XSD 模式的关系



文档模型数据结构代码生成

xgen: 基于模式描述定义为强类型语言生成通用语言数据结构代码



xgen: 基于 XSD 的数据结构代码辅助生成



xgen

Code Generation for XML Schema Definition

安装

```
go get -u -v github.com/xuri/xgen/cmd/...
```

使用

```
$ xgen -i /path/to/your/xsd -o /path/to/your/output -l Go
```

```
$ xgen [<flag> ...] <XSD file or directory> ...
-i <path> Input file path or directory for the XML schema definition
-o <path> Output file path or directory for the generated code
-p          Specify the package name
-l          Specify the language of generated code (Go/C/Java/Rust/TypeScript)
-h          Output this help and exit
-v          Output version and exit
```


示例: 基于 XSD 生成的代码

```
<schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="http://example.org/">
  <simpleType name="myType1">
    <restriction base="base64Binary">
      <length value="10" />
    </restriction>
  </simpleType>
  <complexType name="myType2">
    <simpleContent>
      <extension base="base64Binary">
        <attribute name="length" type="int"/>
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="myType3">
    <simpleContent>
      <extension base="date">
        <attribute name="length" type="int"/>
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="myType4">
    <sequence>
      <element name="title" type="string"/>
      <element name="blob" type="base64Binary"/>
      <element name="timestamp" type="dateTime"/>
    </sequence>
  </complexType>
  <simpleType name="myType5">
    <restriction base="gDay"/>
  </simpleType>
</schema>
```



```
// Code generated by xgen. DO NOT EDIT.
package schema

import "encoding/xml"

// MyType1 ...
type MyType1 string

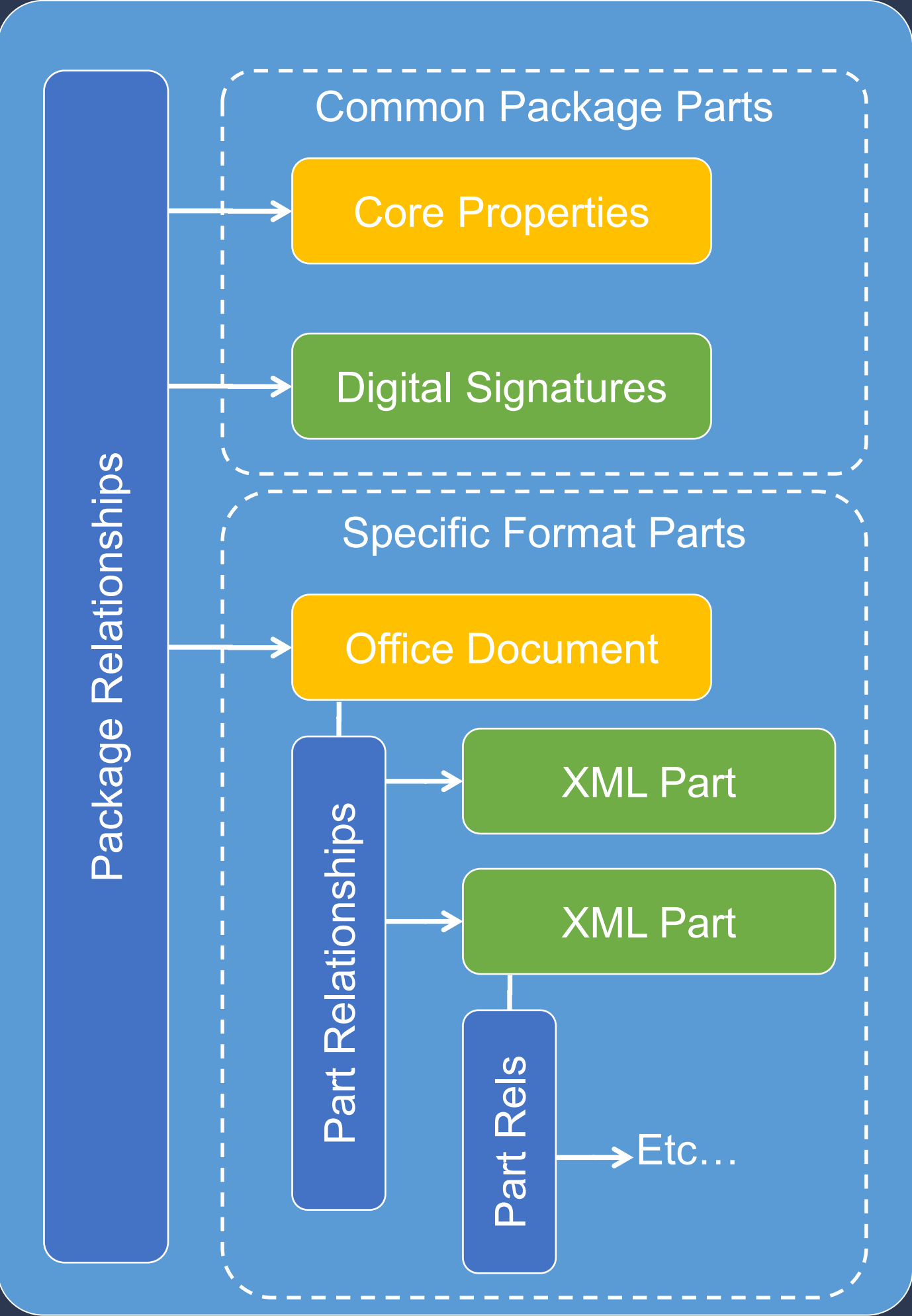
// MyType2 ...
type MyType2 struct {
  XMLName xml.Name `xml:"myType2"`
  LengthAttr int `xml:"length,attr,omitempty"`
  Value string `xml:",chardata"`
}

// MyType3 ...
type MyType3 struct {
  XMLName xml.Name `xml:"myType3"`
  LengthAttr int `xml:"length,attr,omitempty"`
  Value string `xml:",chardata"`
}

// MyType4 ...
type MyType4 struct {
  XMLName xml.Name `xml:"myType4"`
  Title string `xml:"title"`
  Blob string `xml:"blob"`
  Timestamp string `xml:"timestamp"`
}

// MyType5 ...
type MyType5 string
```

基于文档内存模型的运行时操作



	A	B	C
1			
2		123	
3			
4			

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<worksheet
  xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006">
  <dimension ref="B2"/>
  <sheetViews>
    <sheetView tabSelected="1" workbookViewId="0" />
  </sheetViews>
  <sheetFormatPr baseColWidth="10" defaultRowHeight="16" />
  <sheetData>
    <row r="2">
      <c r="B2">
        <v>123</v>
      </c>
    </row>
  </sheetData>
  <pageMargins left="0.7" right="0.7" />
</worksheet>
```

电子表格文档中的数字格式

	类别	选项	格式化结果
45174.43403	常规		45174.43403
	数值	使用千分位分隔符 (,)	45,174.43
	货币	负数	\$45,174.43403
	会计专用	小数位数	\$45,174.43
	日期	符号	Tuesday, September 5, 2023
	时间	日历类型	10:25
	百分比	语言 (位置)	4517443.40%
	分数		45174 3/7
	科学计数	类型	4.52E+04
	文本		45174.43403
	特殊		00004-5174
	自定义	[\$-ja-JP-x-gannen,80]gggee"年"m"月"d"日" dddd;@	令和05年9月5日 火曜日

如何应用数字格式?

-123.456



格式化正数的值

零的格式

格式化负数的值

格式化文本内容

```
#,###.00_);[Red](#.00);0.00;"Sales"@
```

NFP (Number Format Parser)
<https://github.com/xuri/nfp>

```
ps := nfp.NumberFormatParser()
ps.Parse(`#,###.00_`); [Red](#.00); 0.00; "Sales"@`
println(ps.PrettyPrint())
```

```
<Positive>
    # <HashPlaceholder>
    , <ThousandsSeparator>
    ### <HashPlaceholder>
    . <DecimalPoint>
    00 <ZeroPlaceholder>
<Negative>
    Red <Color>
    ( <Literal>
    # <HashPlaceholder>
    . <DecimalPoint>
    00 <ZeroPlaceholder>
    ) <Literal>
<Zero>
    0 <ZeroPlaceholder>
    . <DecimalPoint>
    00 <ZeroPlaceholder>
<Text>
    Sales <Literal>
    @ <TextPlaceholder>
```

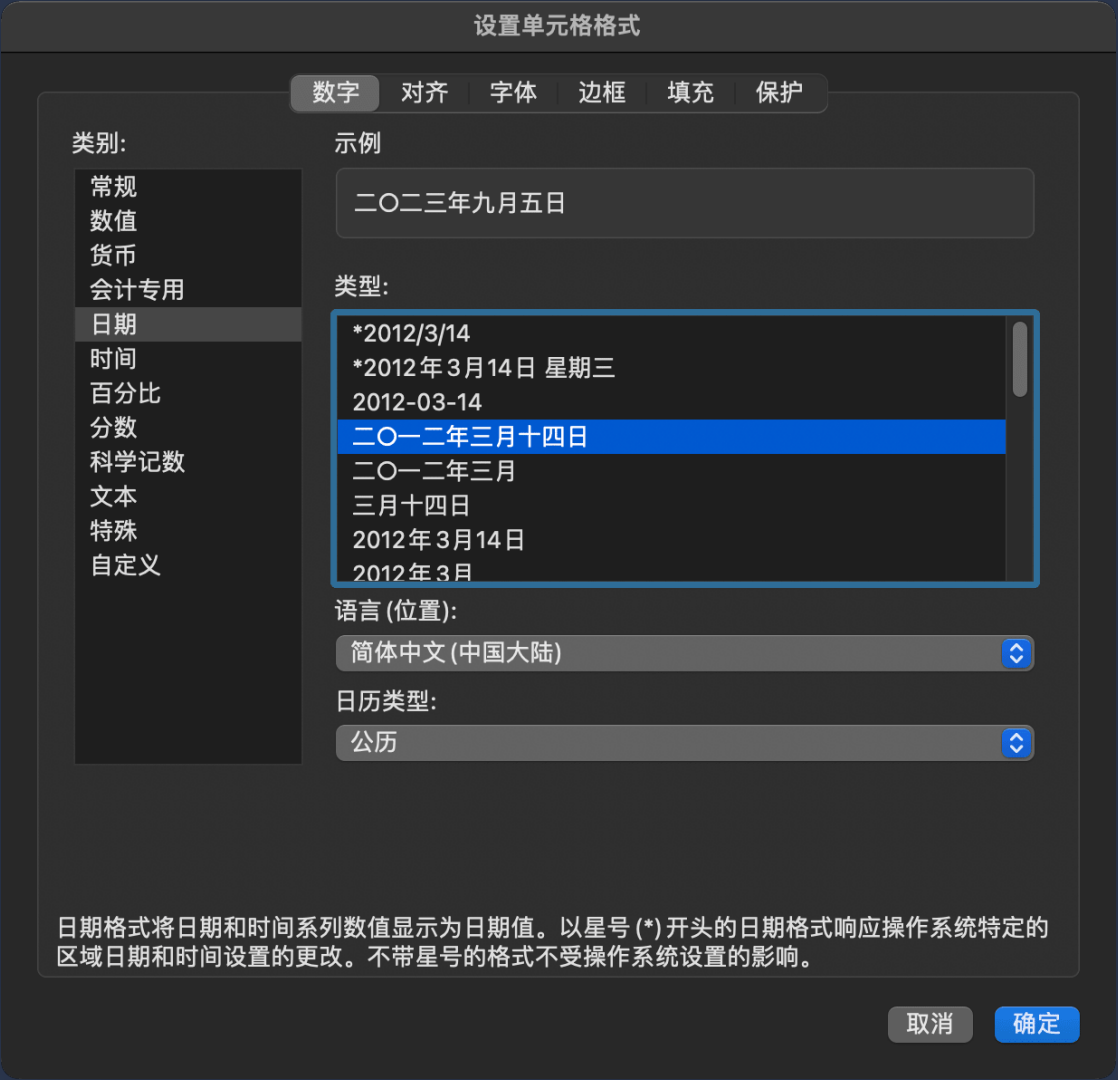

实验评估: 时间与日期类型的数字格式

电子表格文档中的预设数字格式与预设语言地区数字格式 ID 区间范围:
27 ~ 36, 50 ~ 81, 自定义数字格式 ID 的起始值为 164

ID	Win Office 07 OS 国家或地区: 美国 OS 区域格式: 中文(简体,中国) Office 编辑语言: 英文(英语,美国)	Win Office 2021 OS 国家或地区: 美国 OS 区域格式: 中文(简体,中国) Office 编辑语言: 中文(简体,中国)	mac Office 16.3 OS 国家或地区: 美国 OS 区域格式: 美国 Office 编辑语言: 中文(繁体,中国台湾)	mac Office 16.3 OS 国家或地区: 美国 OS 区域格式: 美国 Office 编辑语言: 英文(英语,美国)	Win Office 07 OS 国家或地区: 美国 OS 区域格式: 中文(繁体,中国台湾) Office 编辑语言: 中文(繁体,中国台湾)	Win Office 07 OS 国家或地区: 美国 OS 区域格式: 中文(繁体,中国台湾) Office 编辑语言: 中文(简体,中国)	Win Office 07 OS 国家或地区: 美国 OS 区域格式: 中文(繁体,中国台湾) Office 编辑语言: 中文(简体,中国)	mac Office 16.3 OS 国家或地区: 美国 OS 区域格式: 美国 Office 编辑语言: 中文(简体,中国)	mac Office 16.3 OS 国家或地区: 美国 OS 区域格式: 美国 Office 编辑语言: 日本語	mac Office 16.3 OS 国家或地区: 美国 OS 区域格式: 美国 Office 编辑语言: 泰语	Win Office 07 OS 国家或地区: 美国 OS 区域格式: 英文(英语,美国) Office 编辑语言: 英文(英语,美国)	Win Office 07 OS 国家或地区: 美国 OS 区域格式: 中文(繁体,中国台湾) Office 编辑语言: 日本語	Win Office 07 OS 国家或地区: 日本語 OS 区域格式: 日本語 Office 编辑语言: 日本語	mac Office 16.3 OS 国家或地区: 美国 OS 区域格式: 美国 Office 编辑语言: 韩国语	Win Office 07 OS 国家或地区: 韩国语 OS 区域格式: 韩国语 Office 编辑语言: 韩国语
LDate	yyyy'年'M'月'd'日'	yyyy'年'M'月'd'日'			yyyy'年'M'月'd'日'	yyyy'年'M'月'd'日'	yyyy'年'MM'月'dd'日'				dddd, MMMM d, yyyy	yyyy'年'M'月'd'日'	yyyy'年'M'月'd'日'		dddd, MMMM d, yyyy
LTime	H:mm:ss	H:mm:ss			tt hh:mm:ss	tt hh:mm:ss	HH:mm:ss				H:mm:ss tt	tt hh:mm:ss	H:mm:ss		H:mm:ss tt
SDate	yyyy/M/d	yyyy/M/d			yyyy/M/d	yyyy/M/d	yyyy-MM-dd				M/d/yyyy	yyyy/M/d	yyyy/MM/dd		M/d/yyyy
STime	H:mm	H:mm			tt hh:mm	tt hh:mm	HH:mm				H:mm tt	tt hh:mm	H:mm		H:mm tt
27	2023/8/24	2023年8月	2023/8/24	2023年8月24日	8/24/23	112/8/24	2023年8月	2023年8月	2023年8月	8.24.R5	8/24/23	8/24/2023	H35.08.24	8月 24日 2023年	8月 24日 2023年
28	2023/8/24	8月24日	2023年8月24日	8/24/23	8/24/23	112年8月24日	8月24日	8月24日	8月24日	8月24日 令和5年	8/24/23	8/24/2023	平成35年8月24日	平成35年8月24日	8/24
29	2023/8/24	8月24日	2023年8月24日	8/24/23	8/24/23	112年8月24日	8月24日	8月24日	8月24日	8月24日 令和5年	8/24/23	8/24/2023	平成35年8月24日	平成35年8月24日	8/24
30	2023/8/24	8/24/23	8/24/23	8/24/23	8/24/23	8/24/23	8/24/23	8/24/23	8/24/23	8/24/23	8/24/23	8/24/2023	08-24-23	8/24/23	8/24/23
31	2023/8/24	2023年8月24日	2023年8月24日	8/24/23	8/24/23	2023年8月24日	2023年8月24日	2023年8月24日	2023年8月24日	8月24日 2023年	8/24/23	8/24/2023	2023年8月24日	2023年8月24日	8월 24일 2023년
32	0:00:00	0时00分	0時00分	00:00:00	00:00:00	00時00分	00时00分	00时00分	00时00分	0時00分	00:00:00	0:00:00	00時00分	00時00分	00시 00분
33	0:00:00	0时00分00秒	0時00分00秒	00:00:00	00:00:00	00時00分00秒	00时00分00秒	00时00分00秒	00时00分00秒	0時00分00秒	00:00:00	0:00:00	00時00分00秒	00時00分00秒	0시 00분 00초
34	0:00:00	上午12时00分	上午12時00分	00:00:00	00:00:00	上午12時00分	上午12时00分	上午12时00分	上午12时00分	2023年8月	00:00:00	0:00:00	2023年8月	2023年8月	8-24-23
35	0:00:00	上午12时00分00秒	上午12時00分00秒	00:00:00	00:00:00	上午12時00分00秒	上午12时00分00秒	上午12时00分00秒	上午12时00分00秒	8月24日	00:00:00	0:00:00	8月24日	8月24日	8/24/2023
36	2023/8/24	2023年8月	2023/8/24	8/24/23	8/24/23	112/8/24	2023年8月	2023年8月	2023年8月	8.24.R5	8/24/23	8/24/2023	H35.08.24	H35.8.24	8月 24日 2023年
50	2023/8/24	2023年8月	2023/8/24	8/24/23	8/24/23	112/8/24	2023年8月	2023年8月	2023年8月	8.24.R5	8/24/23	8/24/2023	H35.08.24	H35.8.24	8月 24日 2023年
51	2023/8/24	8月24日	2023年8月24日	8/24/23	8/24/23	112年8月24日	8月24日	8月24日	8月24日	8月24日 令和5年	8/24/23	8/24/2023	平成35年8月24日	平成35年8月24日	8/24
52	2023/8/24	2023年8月	上午12時00分	8/24/23	8/24/23	上午12時00分	2023年8月	2023年8月	2023年8月	2023年8月	8/24/23	8/24/2023	2023年8月	2023年8月	8-24-23
53	2023/8/24	8月24日	上午12時00分00秒	8/24/23	8/24/23	上午12時00分00秒	8月24日	8月24日	8月24日	8月24日	8/24/23	8/24/2023	8月24日	8月24日	8/24/2023
54	2023/8/24	8月24日	2023年8月24日	8/24/23	8/24/23	112年8月24日	8月24日	8月24日	8月24日	8月24日 令和5年	8/24/23	8/24/2023	平成35年8月24日	平成35年8月24日	8/24
55	2023/8/24	上午12时00分	上午12時00分	8/24/23	8/24/23	上午12時00分	上午12时00分	上午12时00分	上午12时00分	2023年8月	8/24/23	8/24/2023	2023年8月	2023年8月	8-24-23
56	2023/8/24	上午12时00分00秒	上午12時00分00秒	8/24/23	8/24/23	上午12時00分00秒	上午12时00分00秒	上午12时00分00秒	上午12时00分00秒	8月24日	8/24/23	8/24/2023	8月24日	8月24日	8/24/2023
57	2023/8/24	2023年8月	112/8/24	8/24/23	8/24/23	2023年8月	2023年8月	2023年8月	2023年8月	8.24.R5	8/24/23	8/24/2023	H35.08.24	H35.8.24	8月 24日 2023年
58	2023/8/24	8月24日	2023年8月24日	8/24/23	8/24/23	112年8月24日	8月24日	8月24日	8月24日	8月24日 令和5年	8/24/23	8/24/2023	平成35年8月24日	平成35年8月24日	8/24
59	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162
60	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00	45162.00
61	45,162	45,162	45,162	45,162	45,162	45,162	45,162	45,162	45,162	45,162	45,162	45,162	45,162	45,162	45,162
62	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00	45,162.00
63	¥45,162	¥45,162	\$45,162	\$45,162	\$45,162	NT\$45,162	NT\$45,162	¥45,162	¥45,162	\$45,162	\$45,162	\$45,162	NT\$45,162	¥45,162	¥45,162
64	¥45,162	¥45,162	\$45,162	\$45,162	\$45,162	NT\$45,162	NT\$45,162	¥45,162	¥45,162	\$45,162	\$45,162	\$45,162	NT\$45,162	¥45,162	¥45,162
65	¥45,162.00	¥45,162.00	\$45,162.00	\$45,162.00	\$45,162.00	NT\$45,162.00	NT\$45,162.00	¥45,162.00	¥45,162.00	\$45,162.00	\$45,162.00	\$45,162.00	NT\$45,162.00	¥45,162.00	¥45,162.00
66	¥45,162.00	¥45,162.00	\$45,162.00	\$45,162.00	\$45,162.00	NT\$45,162.00	NT\$45,162.00	¥45,162.00	¥45,162.00	\$45,162.00	\$45,162.00	\$45,162.00	NT\$45,162.00	¥45,162.00	¥45,162.00
67	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%	4516200%
68	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%	4516200.00%
69	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162
70	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162	45162
71	2023/8/24	2023/8/24	2023/8/24	8/24/23	8/24/23	2023/8/24	2023-08-24	2023/8/24	2023-08-24	2023/8/24	8/24/2023	8/24/2023	2023-08-24	2023/8/24	8/24/23
72	2023/8/24	2023/8/24	2023/8/24	8/24/23	8/24/23	2023/8/24	2023-08-24	2023/8/24	2023-08-24	2023/8/24	8/24/2023	8/24/2023	2023-08-24	2023/8/24	8/24/23
73	24-8月-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23	24-Aug-23
74	24-8月	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug	24-Aug
75	8月-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23	Aug-23
76	0:00	0:00	0:00	00:00	00:00	00:00	00:00	00:00	00:00	0:00	00:00:00	0:00	00:00	0:00	0:00
77	0:00:00	0:00:00	0:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	0:00:00	00:00:00	0:00:00	00:00:00	0:00:00	0:00:00
78	2023/8/24 0:00	2023/8/24 0:00	2023/8/24 0:00	8/24/23 00:00	2023/8/24 00:00	2023/8/24 00:00	2023-08-24 00:00	2023/8/24 00:00	2023-08-24 00:00	8/24/2023 0:00	00:00:00	8/24/2023 0:00	2023-08-24 00:00	2023/8/24 0:00	8/24/23 00:00
79	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00	00:00
80	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00	1083888:00:00
81	00:00.0	00:00.0	00:00.0	00:00.0	00:00.0	00:00.0	00:00.0	00:00.0	00:00.0	00:00.0	8/24/66		00:00.0	00:00.0	00:00.0

数字格式: 国际化 & 本地化

45174.43403



2023/09/05
2023-09-05
2023年9月5日
二〇二三年九月五日

...

Excelize 对时间与日期类型数字格式的解决方案

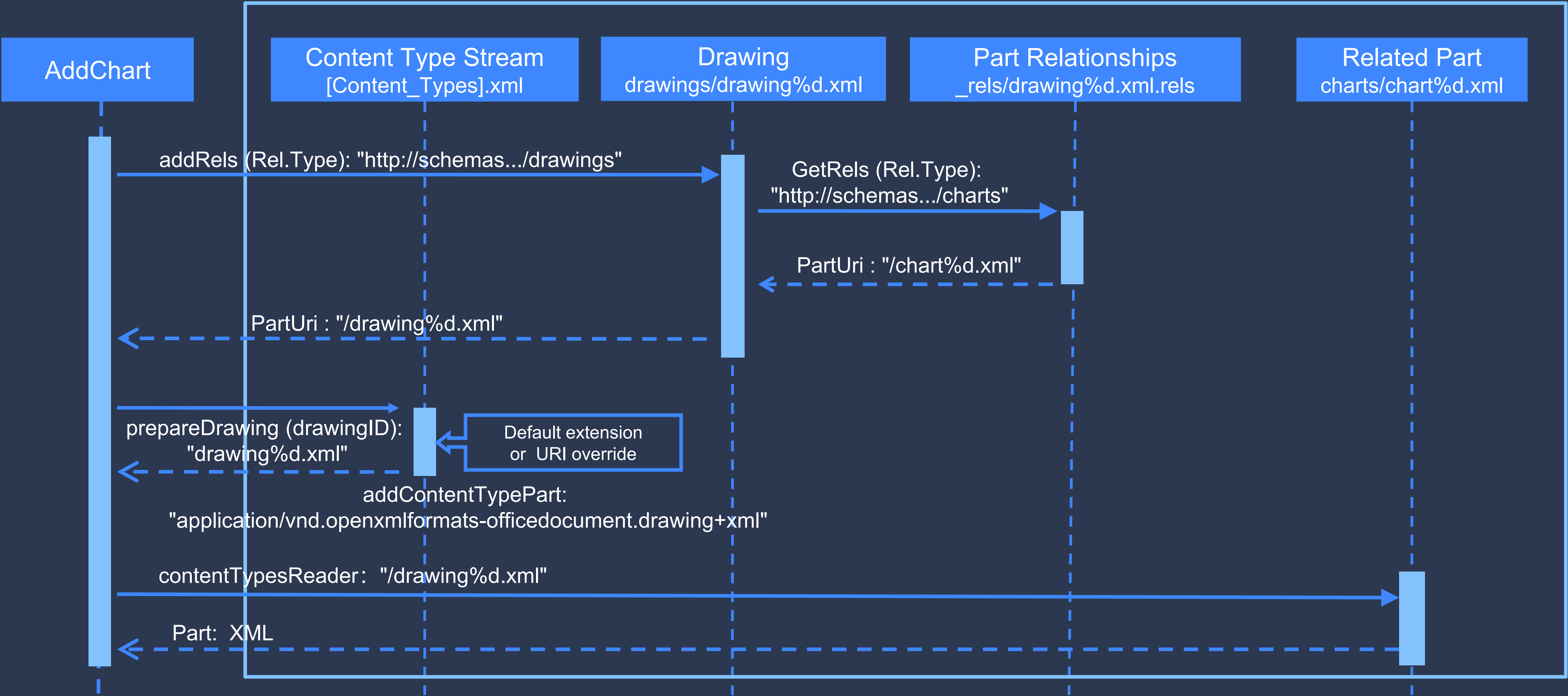
参考标准 [MS-LCID]: Windows Language Code Identifier Reference

Excelize 已覆盖包含超过 810 个语言地区标识的时间与日期类型数字格式

```
f, err := excelize.OpenFile("Book1.xlsx",
    excelize.Options{
        CultureInfo:    excelize.CultureNameZhCN,
        ShortDatePattern: "dd/mm/yy",
        LongDatePattern: "yyyy-mm-dd",
        LongTimePattern: "hh:mm:ss",
    },
)
```

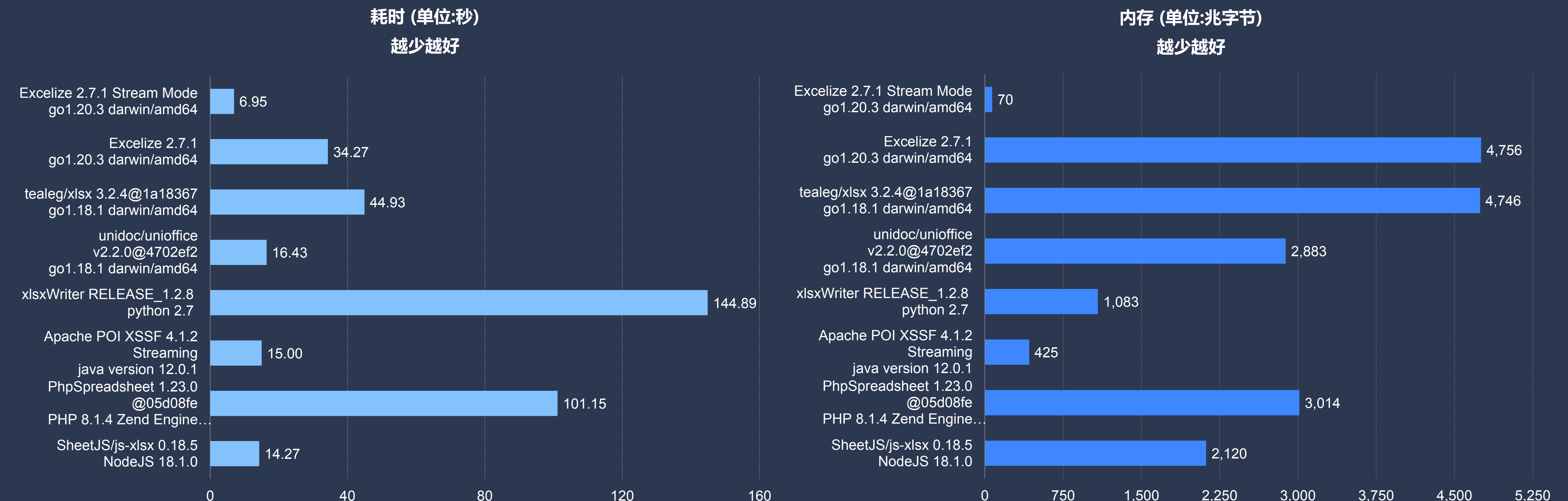


架构设计: 图形处理系统



性能表现

下图展示了 Go, Python, Java, PHP 和 NodeJS 语言中典型 Excel 开源基础库，基于普通个人计算机 (2.6 GHz 6-Core Intel Core i7, 16 GB 2667 MHz DDR4, 500GB SSD, macOS Monterey 12.3.1) 生成 50 列 102400 行纯文本单元格的性能表现

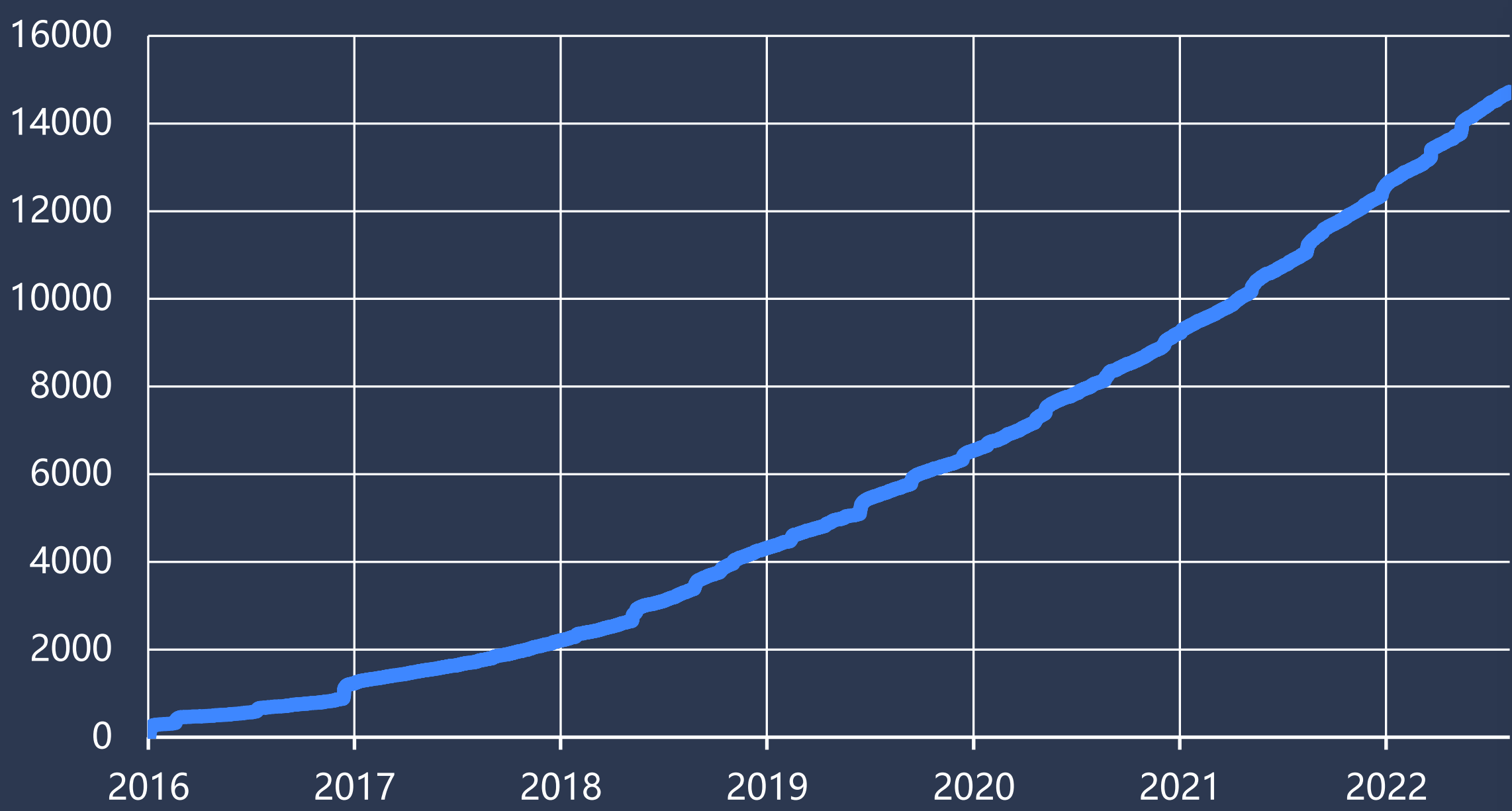


测试脚本: <https://github.com/xuri/excelize-benchmark>

函数性能: <https://xuri.me/excelize/en/performance.html>

开源数据洞察

Excelize GitHub Star 趋势



Excelize 全球范围内贡献者分布



* 截至 2023 年 9 月，GitHub Star 1.5 万，有超过 50 个国家和地区的 400 多名贡献者，其中有 170 余人参与了代码贡献

WebAssembly 跨语言调用

```
npm install --save excelize-wasm
```

```
const { init } = require('excelize-wasm');  
const fs = require('fs');
```

```
init('./node_modules/excelize-wasm/excelize.wasm.gz').then((excelize) => {  
  const f = excelize.NewFile();  
  const { index } = f.NewSheet('Sheet2');  
  f.SetCellValue('Sheet2', 'A2', 'Hello world.');// 单元格赋值  
  f.SetCellValue('Sheet1', 'B2', 100);  
  f.SetActiveSheet(index); // 设置活动工作表  
  const { buffer, error } = f.WriteToBuffer();  
  if (error) {  
    console.log(error);  
    return;  
  }  
  fs.writeFile('Book1.xlsx', buffer, 'binary', (error) => {  
    if (error) {  
      console.log(error);  
    }  
  });  
});
```


总结展望

功能规划

更多接口将提供并发安全支持

对包含大规模数据文档读写性能的持续优化

公式计算引擎增加更多计算函数

复杂条件格式的设置、数据预测支持

灵活控制图表样式

数据透视图、切片器功能

更加丰富的数字格式表达式解析

提供更多可用流式读写功能

THANKS

—
软件正在重新定义世界

Software Is Redefining The World



Excelize 续日



GitHub 代码库