

年中工作总结及未来展望-刘树桂

数据治理

实时数仓-数据总线

规则引擎

数据治理

数仓三部曲：数据接入/模型设计/数据开发

痛点

- 敏捷开发模式，元数据变更频繁，需手动处理，耗费大量人力
- 作业任务繁多，数据接入无法核实准确性
- Dataworks功能尚不完善（如调度问题），使用存在不便

	A	B	C	D	E	F	G	H	I
1	bus_line	module_name	table_schema_name	source_table_name	column_name	target_table_name	result	create_time	update_time
2	bas	ad	ad_service	master_ad_info	is_group	ods_bas_ad_master_ad_info	column missing	2021-08-05 17:18:08	W
3	bas	ad	ad_service	user_ad_info	show_time_limit	ods_bas_ad_user_ad_info	column missing	2021-08-04 16:41:59	W
4	bas	orderconf	order_config_service	serve	serve_public_config_ids	ods_bas_orderconf_serve	column missing	2021-07-30 21:24:36	W
5	bas	orderrefund	order_refund_service	order_refund_extra	real_reward_price	ods_bas_orderrefund_extra	column missing	2021-07-28 22:22:19	W
6	mst	order	master_order_service	order_extra_data	expect_door_in_end_date	ods_mst_order_extra_data_inc	column missing	2021-07-30 23:49:08	W
7	usr	userbase	user_service	face_verify_result	client_type	ods_usr_userbase_face_verify_result	column missing	2021-08-04 16:51:38	W
8	bas	address	base_address_service	hot_city	skip	ods_bas_address_hot_city	table missing	2021-07-28 11:42:31	W
9	bas	address	base_address_service	tencent_district	skip	ods_bas_address_tencent_district	table missing	2021-07-28 11:42:31	W
10	bas	orderconf	order_config_service	adjust_fee_reason	skip	ods_bas_orderconf_adjust_fee_reason	table missing	2021-07-15 04:12:30	W
11	bas	orderconf	order_config_service	serve_view_classify_ocs	skip	ods_bas_orderconf_serve_view_classify_ocs	table missing	2021-07-31 17:00:04	W
12	bas	orderconf	order_config_service	serve_view_classify_ocs_snapshot	skip	ods_bas_orderconf_serve_view_classify_ocs_snapshot	table missing	2021-07-29 20:56:26	W
13	bas	orderconf	order_config_service	temporary	skip	ods_bas_orderconf_temporary	table missing	2021-07-29 21:16:50	W
14	bas	voucher	voucher_service	sys_sequence	skip	ods_bas_voucher_sys_sequence	table missing	2021-07-30 22:04:00	W
15	etp	etp	enterprise_service	subaccount_scheduling	skip	ods_etp_etp_subaccount_scheduling	table missing	2021-08-04 18:04:51	W
16	etp	order	enterprise_order_service	order_demand_extra_fee_apply_image	skip	ods_etp_order_demand_extra_fee_apply_image	table missing	2021-07-14 17:44:40	W
17	etp	order	enterprise_order_service	order_demand_extra_fee_apply_log	skip	ods_etp_order_demand_extra_fee_apply_log	table missing	2021-07-14 17:44:40	W
18	mst	contract	master_contract_service	contract_technique_info	skip	ods_mst_contract_technique_info	table missing	2021-07-13 14:31:51	W
19	mst	contract	master_contract_service	platform_contract_price_log	skip	ods_mst_contract_platform_contract_price_log	table missing	2021-07-14 18:31:52	W
20	mst	contract	master_contract_service	recruit_config	skip	ods_mst_contract_recruit_config	table missing	2021-07-14 18:31:51	W
21	mst	order	master_order_service	add_follow_sub_master_log	skip	ods_mst_order_add_follow_sub_master_log	table missing	2021-07-14 18:31:52	W
22	mst	order	master_order_service	order_limit_offer_set_v2	skip	ods_mst_order_limit_offer_set_v2	table missing	2021-07-29 21:08:12	W
23	usr	activity	activity	toc_user_heartbeat	skip	ods_usr_activity_toc_user_heartbeat	table missing	2021-07-29 21:17:16	W
24	usr	activity	activity	toc_user_heartbeat_collect	skip	ods_usr_activity_toc_user_heartbeat_collect	table missing	2021-07-29 21:17:16	W
25	usr	order	user_order_service	order_contact_master_record	skip	ods_usr_order_contact_master_record	table missing	2021-07-29 21:00:32	W

方案与规划

针对痛点，开发监控系统对Dataworks进行功能拓展，保障数据质量，主要功能如下：

- 监听生产环境元数据变更，提供解决方案，实现自动化更新迭代
- 对业务库、数据仓库数据量进行监控，将异常信息通知到钉钉群人工分析处理
- 依据OpenApi，实现开发环境自动调度，避免因调度问题延误数分、测试同事工期

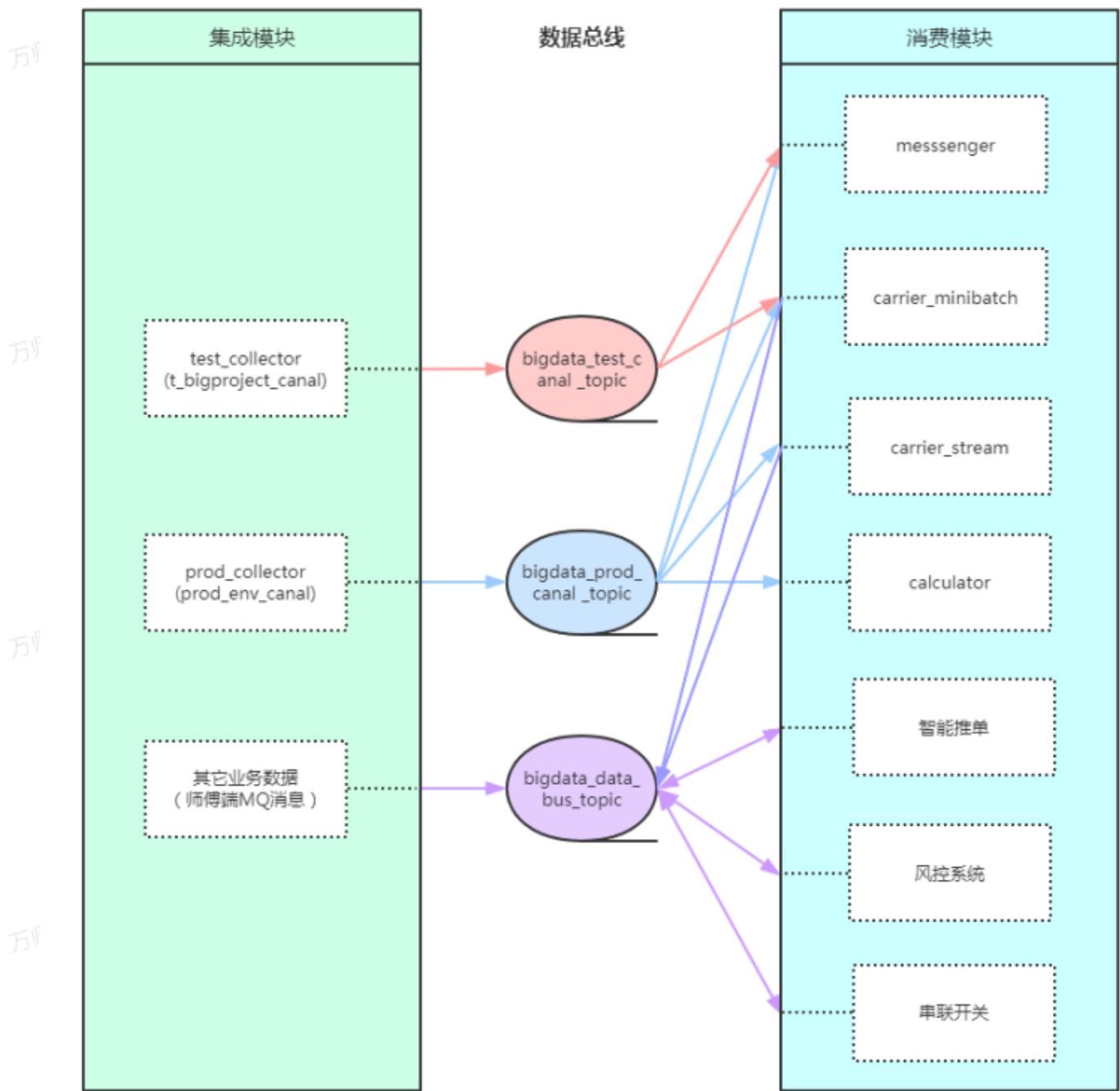
愿景

数仓保质保量，90%以上业务迭代场景可通过自动化调整应对，无需人工干预；

数分同事无需关心数据接入，精力主要集中于业务开发

实时数仓-数据总线

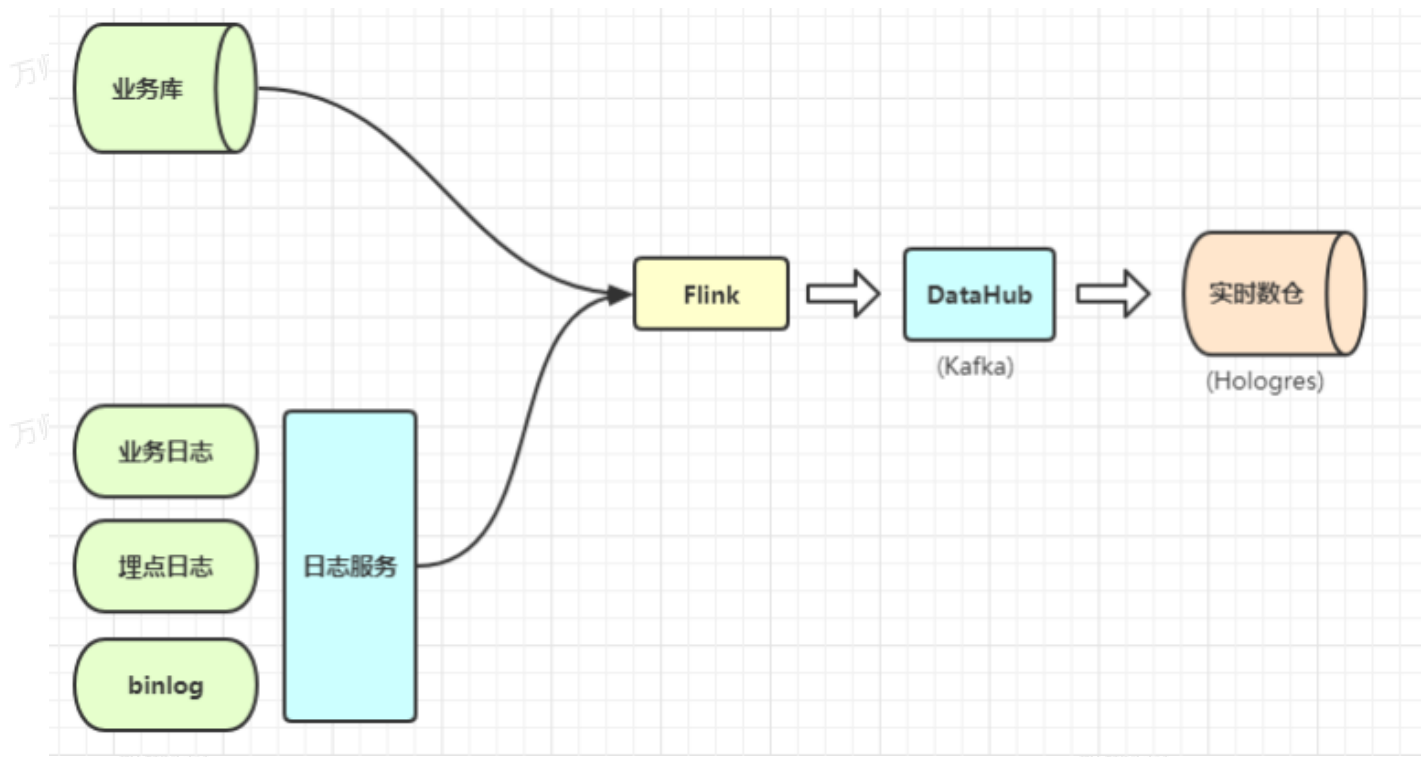
现状：依赖自研中间件，实现数据的实时同步计算，对外部系统提供实时数据服务



痛点

- 性能存在瓶颈，当数据量剧增时，可能会出现同步计算延迟甚至系统宕机，需要人工手动修复
- 规则配置复杂，功能支持存在局限性，维护人力成本较高
- 系统间耦合严重，下游与业务系统交互混乱，没有统一规范
- 数据通用性低。仅在大数据内部使用，不支持其他业务线数据共享

规划及愿景



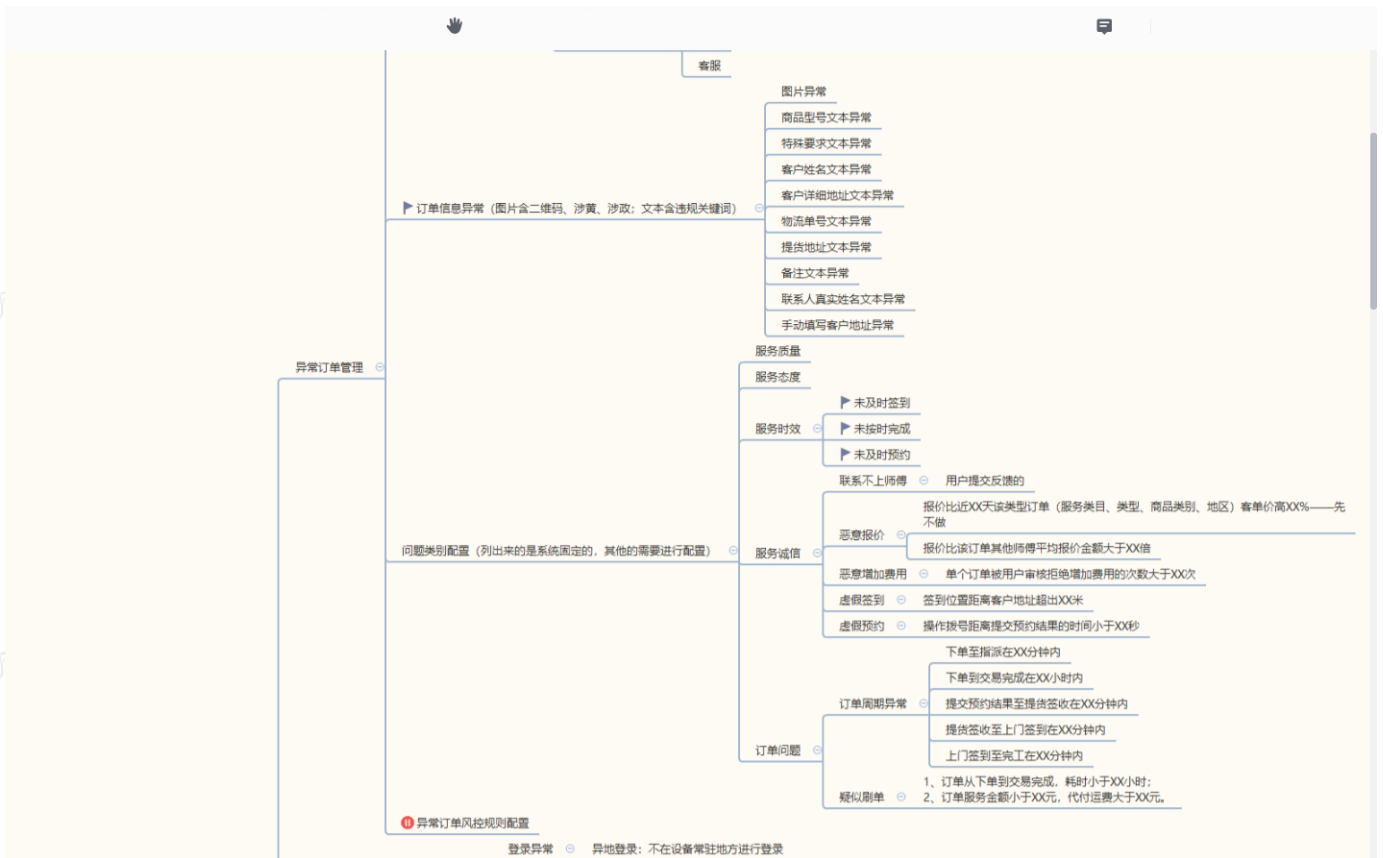
依托于阿里云DataHub（暂定），重新设计新的总线规范，提高数据使用价值：

- 统一数据接入格式，确定数据落地存储形式
- 提供数据消费渠道，增强数据可复用性
- 增设权限控制，保障数据安全
- 按量付费，自动扩容，保证总线稳定性

规则引擎

名词解释：将业务决策从应用程序代码中分离出来，并使用预定义的语义模块编写业务决策

应用场景：应用面较广，如风控系统、智能推单系统、临时取数...等



现有痛点

- 可扩展性低: 新增一个规则配置, 开发需要针对性进行编码, 效率低下
- 沟通成本高: 规则从制定到上线, 需求提出者需与开发人员达成共识, 存在较大的沟通成本
- 上线周期长: 需求从提出到上线, 需要经历需求->设计->编码的过程, 少则几天, 多则数月
- 风险大: 业务规则经常多变, 技术实现也需要频繁调整, 存在较大的上线风险

规划及愿景:

预定9月初项目立项, 以临时取数场景作为切入点, 优先实现规则类规则解析。对Drools开源规则引擎进行二次开发, 基于离线数仓基础指标, 实时解析用户规则配置, 自动发布执行统计任务, 降低人力成本

规则引擎是未来的一个大方向, 可应用于现有绝大多数项目, 例如风控、推单等。规则引擎能够实现业务规则和程序开发解耦, 让产品与开发专注各自的领域, 实现即配即用, 无需等待漫长的上线流程