

서비스 DB 구축 발표

개인 맞춤형 음료 제공 서비스

헬th | 드링크

2019. 08. 07. 3조 천하림, 000, 000

본 문서의 모든 콘텐츠는 저작권법의 보호를 받는 저작물로 별도의 저작권 표시 또는 다른 출처를 명시한 경우를 제외하고는 (주)인실리코젠에 저작권이 있습니다.
저작권 표시 또는 기타 소유권 표시를 삭제해서도 안되며, 당사와의 협의 또는 허락없이 무단 복제, 변경, 배포를 금지합니다.
저작권 관련 문의사항이 있으시면 bc@insilicogen.com으로 연락바랍니다.

© 2019 INSILICOGEN, INC. ALL RIGHTS RESERVED.

INSILICOGEN

www.insilicogen.com

INDEX

01 Purpose

02 BCM

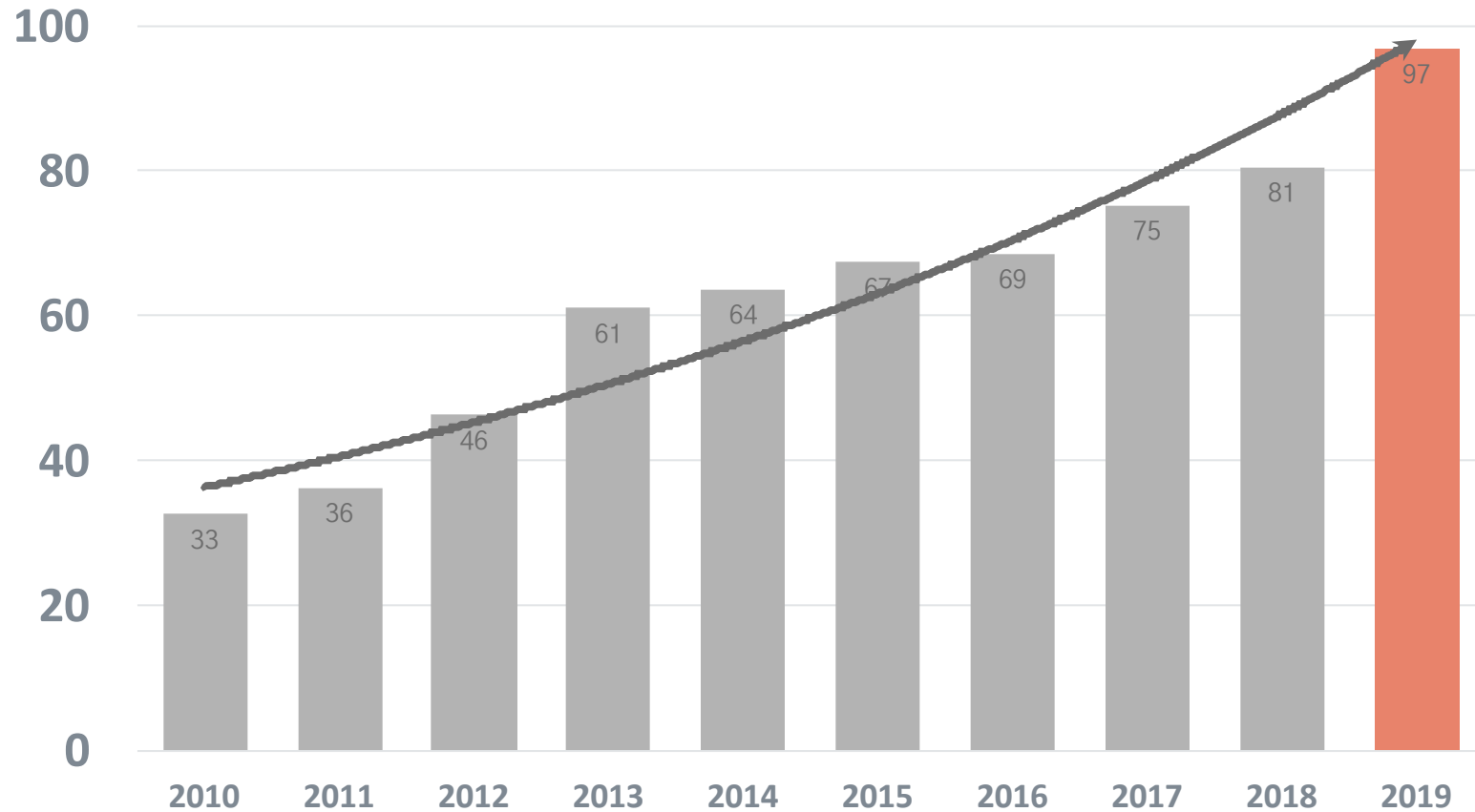
03 DB 구축

04 Prototype

01

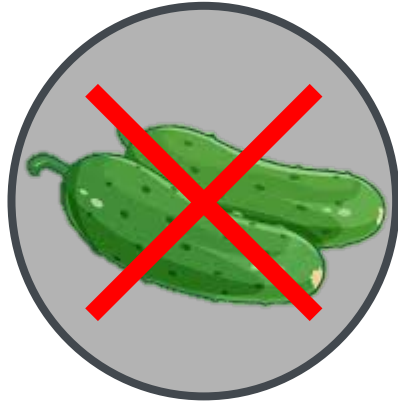
Purpose

건강 음료 관련 검색 빈도 변화

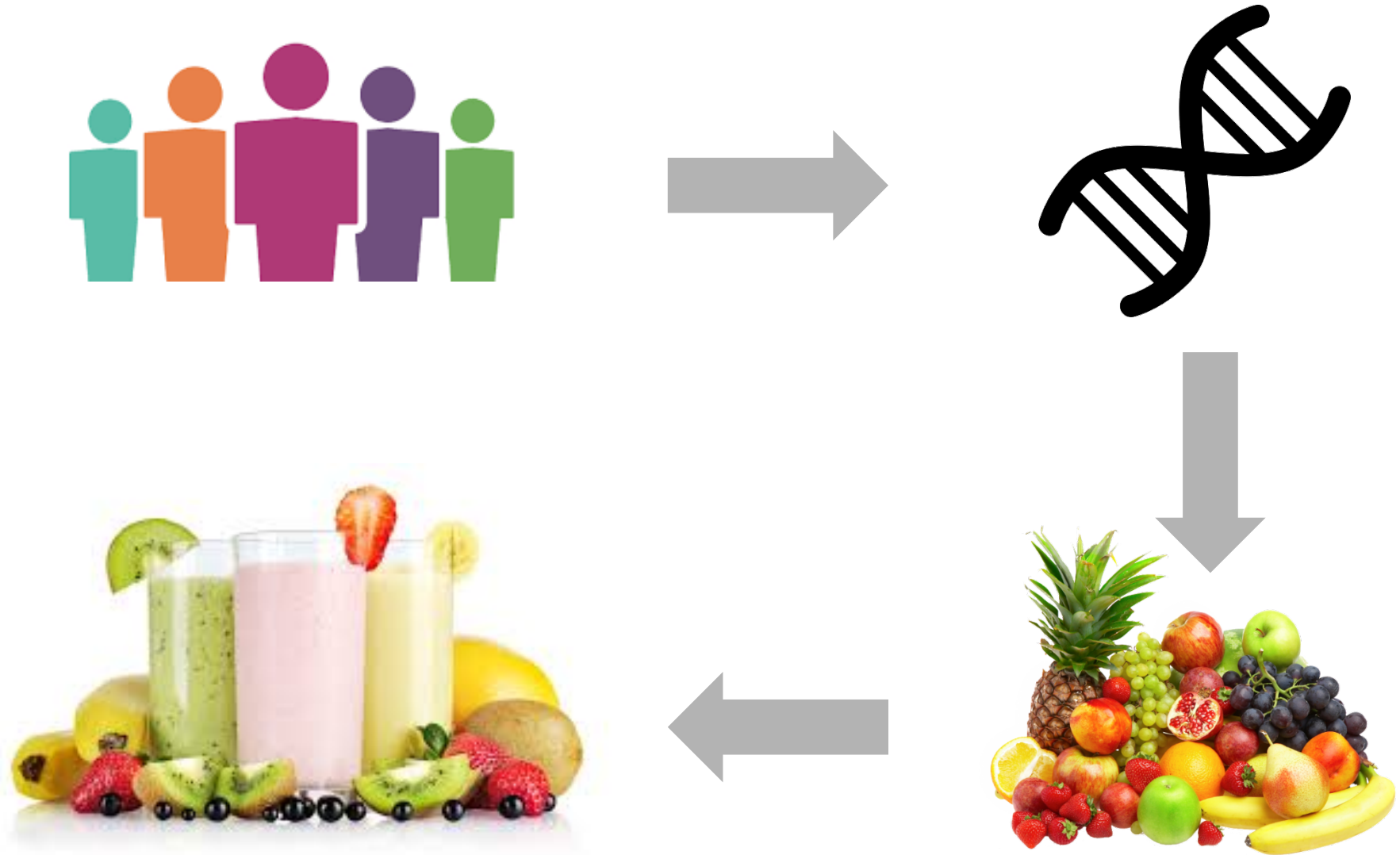


출처 : Google 트렌드

Purpose



Purpose



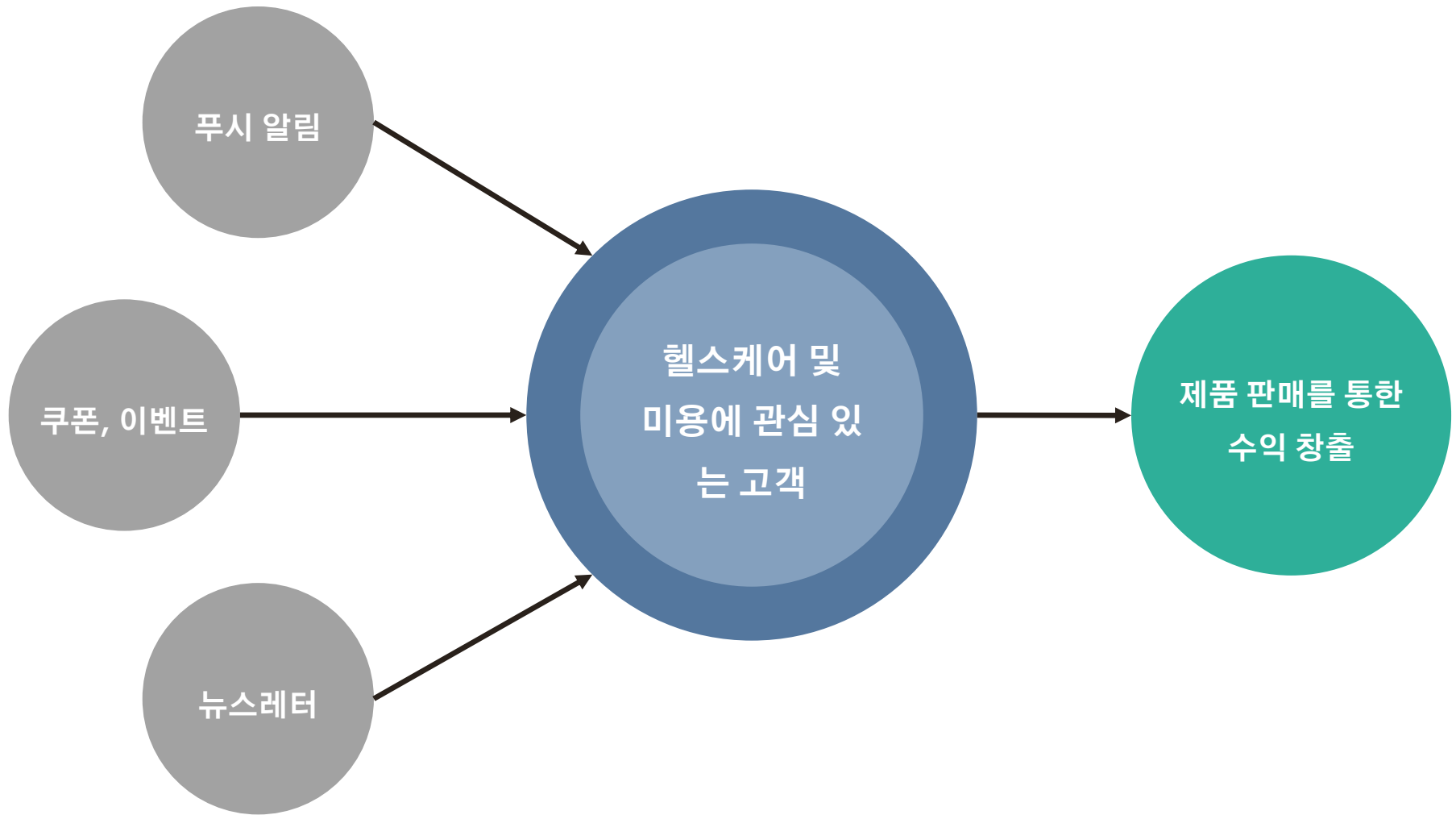
02

BCM

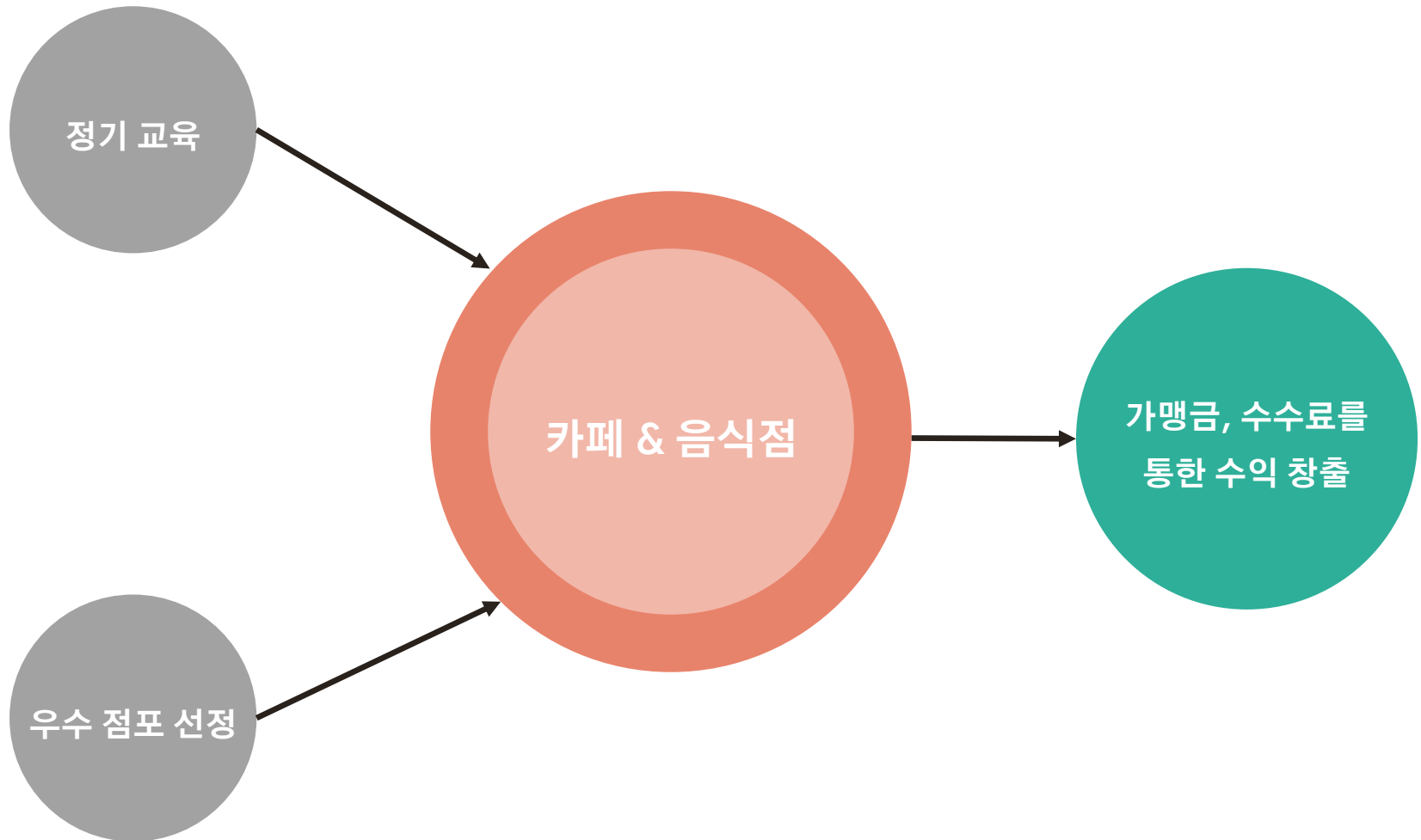
BCM(Business Canvas Model)



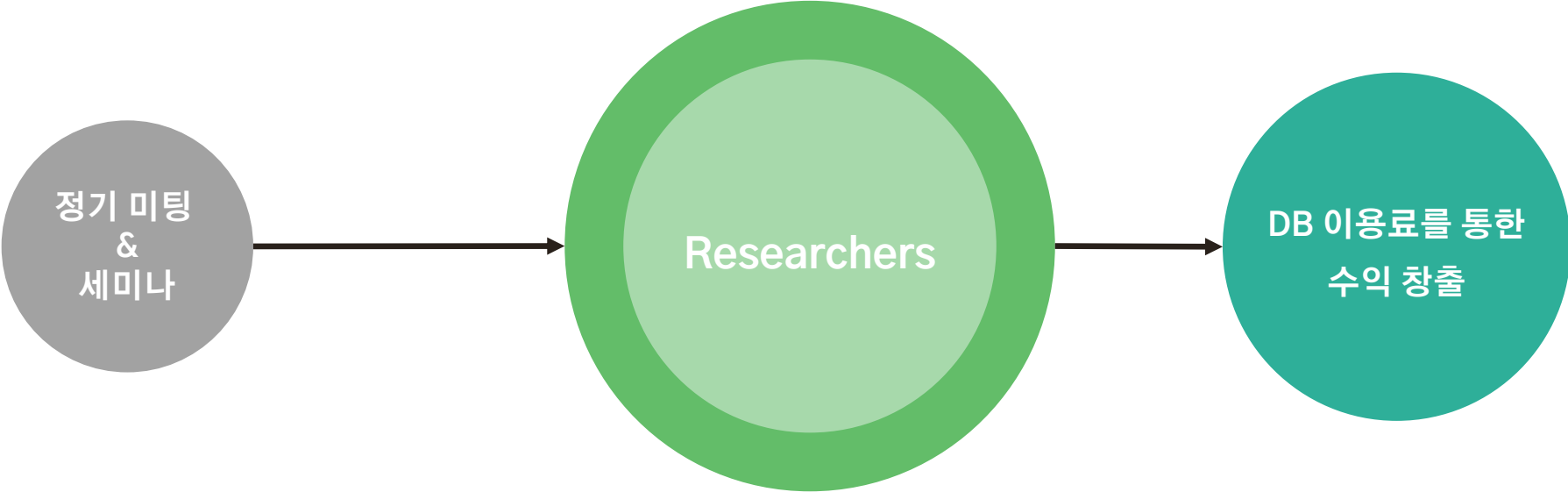
Customer Segments & Revenue Streams



Customer Segments & Revenue Streams



Customer Segments & Revenue Streams

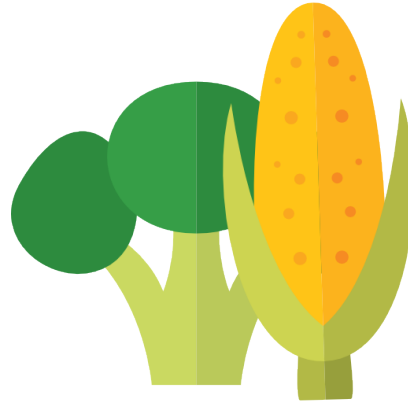


Key Partners



유전자 검사 관련 업체와 협력하여 유전자 검사 키트 수령 및 분석 의뢰

Key Partners



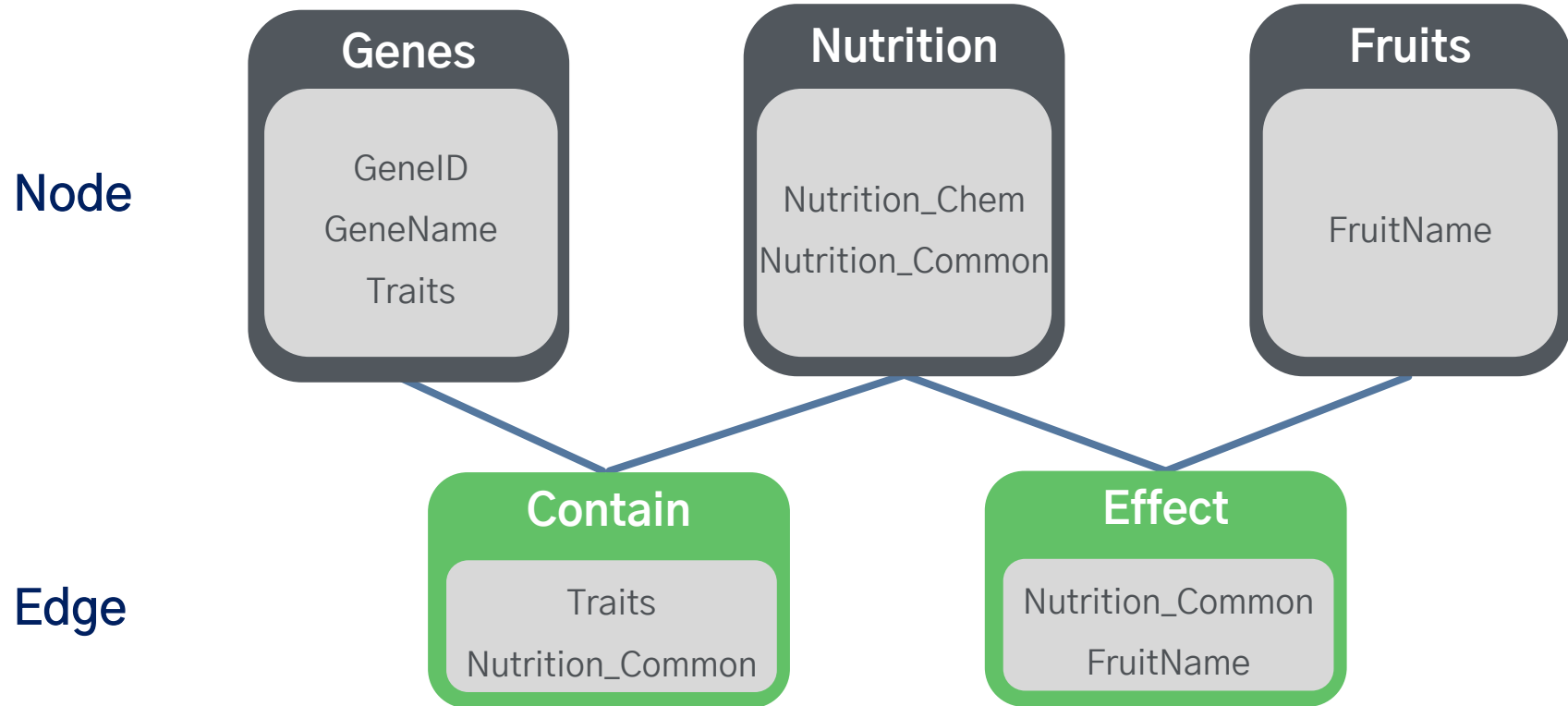
과수원, 농장과 파트너를 맺음으로써 신선한 원재료를 제공받을 수 있다.



택배사와의 계약을 통해 소비자에게 제품 배달 서비스 제공 및 가맹점에서 실시된 유전자검사 키트를 수거 할 수 있다.

03

DB 구축



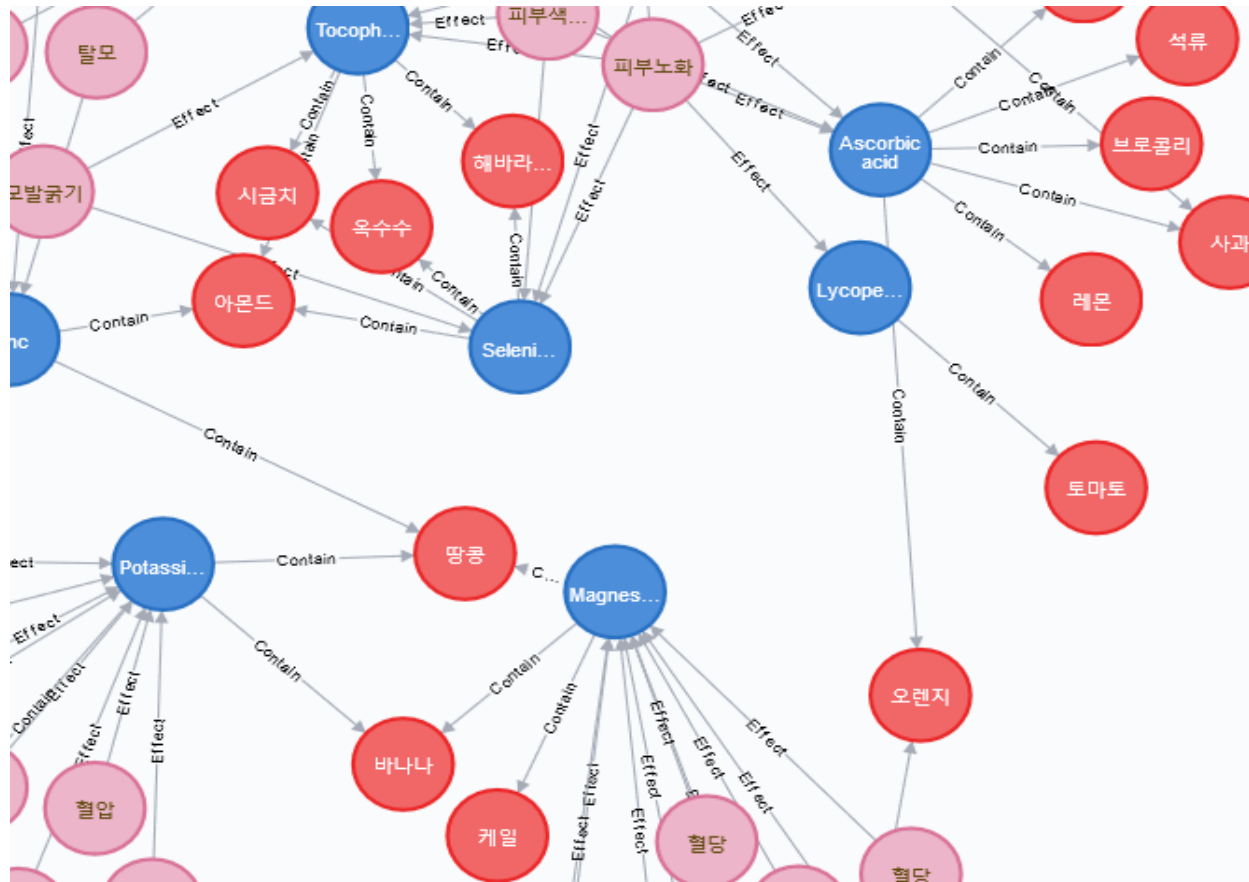
DTC 유전자 검사를 통해 얻은 12가지 특성과 영양성분, 과일 정보를 통해 노드 및 엣지 생성

데이터베이스 구축

	B	C	E	F	H	J	K
1	GeneName	Traits	Nutrition_Chem	Nutrition_Common	FruitName	Traits	Nutrition_Commo
2	AGER	피부노화	Ascorbic acid	Vitamin C	석류	피부노화	Vitamin C
3	OCA2	피부색소침착	Tocopherol	Vitamin E	사과	피부노화	Vitamin E
4	MC1R	피부색소침착	Cartenoids	Vitamin A	오렌지	피부노화	Vitamin A
5	MMP1	피부탄력	Collagen	Collagen	키위	피부노화	Polyphenols
6	AHR	카페인대사	Niacin	Vitamin B	브로콜리	피부노화	Lycopene
7	CYP1A1-CYP1A2	카페인대사	Zinc	Zinc	레몬	피부노화	Flavonoids
8	EDAR	모발굼기	Polyphenols	Polyphenols	당근	피부색소침착	Vitamin C
9	chr20p11	탈모	Flavonoids	Flavonoids	호박	피부색소침착	Vitamin E
10	HLA-DQB1	탈모	Magnesium	Magnesium	망고	피부색소침착	Vitamin A
11	IL2RA	탈모	Cellulose	Fiber	양배추	피부색소침착	Polyphenols
12	SLC23A1	비타민C농도	Selenium	Vitamin E	옥수수	피부색소침착	Flavonoids
13	CDKN2A	혈당	Potassium	Potassium	아몬드	피부탄력	Collagen
14	CDKN2B	혈당	Caffeine	Caffeine	시금치	피부탄력	Vitamin B
15	G6PC2	혈당	Omega-3 fatty acids	n-3 fatty acids	해바라기씨	피부탄력	Zinc
16	GCK	혈당	Biotin	Vitamin H	꽃감	혈당	Magnesium
17	GCKR	혈당	Lycopene	Lycopene	현미	혈당	Fiber
18	GLIS3	혈당	beta-D-galacturonic acid	Pectin	우유	혈당	Chromium
19	MTNR1B	혈당			요거트	혈압	Potassium
20	DGKB-TMEM195	혈당			바나나	체질량지수	Caffeine
21	SLC30A8	혈당			땅콩	콜레스테롤	Pectin
22	NPR3	혈압			아보카도	탈모	Omega-3 fatty ac
23	ATP2B1	혈압			케일	탈모	Vitamin A
24	NT5C2	혈압			파인애플	탈모	Biotin

특정 유전자에 의해 나타나는 Traits과 그 특성에 연관된 영양성분

그리고 그 영양성분이 풍부하게 포함된 과일을 조사 후 데이터베이스에 등록



만들어진 DB를 Neo4j를 이용해 노드와 엣지로 등록하고 Graph DB 작성

04

Prototype

Workflow

체질량지수	중성지방농도	콜레스테롤	혈당	혈압	색소침착
위험	주의	위험	정상	위험	정상
탈모	모발 굵기	피부 노화	피부 탄력	비타민 C농도	카페인 대사
위험	정상	주의	위험	주의	정상



```
$ match (n:Genes)-[r]-(n1:Nutrition) with n, n1 match (n2:Fruits)-[r]-(n1:Nutrition) where n.Traits in ['탈모']
return distinct n.Traits, n1.Nutrition_Common, n2.FruitName ORDER by n.Traits, n1.Nutrition_Common
```

n.Traits	n1.Nutrition_Common	n2.FruitName
탈모	"Vitamin A"	*당근*
탈모	"Vitamin A"	*호박*
탈모	"Vitamin A"	*망고*
탈모	"Vitamin A"	*양배추*
탈모	"Zinc"	*아몬드*
탈모	"Zinc"	*쌍콩*
탈모	"Zinc"	*잣*
탈모	"Zinc"	*캐슈넛*

Started streaming 8 records after 6 ms and completed after 6 ms.



```
http.socket_timeout = 9999

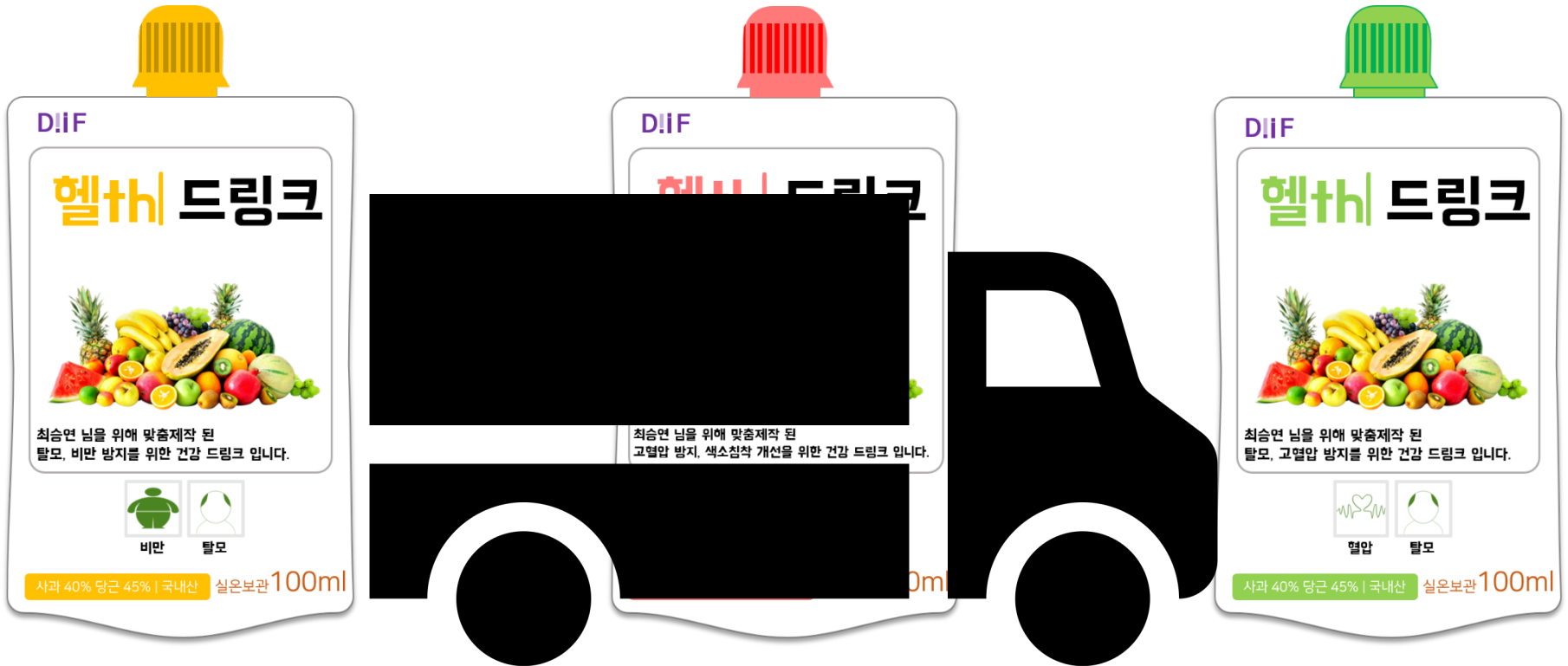
authenticate('localhost:7474', 'neo4j', 'jwllowe1020') ## 비밀번호만 설정해주면 됨
graph = Graph('http://localhost:7474/db/data/')

def query_result(args):
    traits_dict = {1: '피부노화', 2: '피부색소침착', 3: '피부탄력', 4: '카페인대사', 5: '모발굵기', 6: '탈모', 7: '비타민C농도', 8: '혈당', 9: '혈압', 10: '체질량지수', 11: '중성지방농도', 12: '콜레스테롤'}
    a = (input('확인하고 싶은 Traits를 입력해주세요. #n1, 피부노화 #n2, 피부색소침착#n3, 피부탄력#n4, 카페인대사#n5, 모발굵기#n6, 탈모#n7, 비타민C농도#n8, 혈당#n9, 혈압#n10, 체질량지수#n11, 중성지방농도#n12, 콜레스테롤#n'))
    trait_input = a.split()
    trait_num = []
    traits_list = []
    for i in range(len(trait_input)):
        trait_num.append(int(trait_input[i]))
    for j in trait_num:
        traits_list.append(traits_dict[j])
    print('선택하신 특성은 ', traits_list, '입니다.')
    name1 = ','.join(traits_list)

    query = "match (n:Genes)-[r]-(n1:Nutrition) with n, n1 match (n2:Fruits)-[r]-(n1:Nutrition) where n.Traits in ['" + name1 + "'] return distinct n.Traits, n1.Nutrition_Common, n2.FruitName ORDER by n.Traits, n1.Nutrition_Common"
    query1 = "match (n:Genes)-[r]-(n1:Nutrition) with n, n1 match (n2:Fruits)-[r]-(n1:Nutrition) where n.Traits in ['" + name1 + "'] return distinct n1.Nutrition_Common, n2.FruitName ORDER by n1.Nutrition_Common"
    da_result = graph.cypher.execute(query)
    da_result1 = graph.cypher.execute(query1)
    da_record = [record for record in da_result]
    da_record1 = [record for record in da_result1]
    df = pd.DataFrame(da_record)
    df1 = pd.DataFrame(da_record1)
    pd.set_option('display.max_rows', 500)
    df.columns = ['Traits', 'Nutrition', 'Fruit']
    df1.columns = ['Nutrition', 'Fruit']
    df1.to_excel('test1.xlsx')
    return print(df, ' #n1', df1, ' #n2') ##고려님의 견해를 지키기 위해 추천 된 영양소와 재료입니다. ##n1
```



음료 제공(공장제조)



음료 제공(카페 및 음식점)



음료 제공(카페 및 음식점)



건강 보조제는 건강 보조제일 뿐이다.

Thank you!

Contact Info.

OFFICE 경기도 용인시 기흥구 흥덕1로 13 흥덕IT밸리 타워 A동 2901-2903호
EMAIL info@insilicogen.com
PHONE # 031 278 0061
FAX 031 278 0062